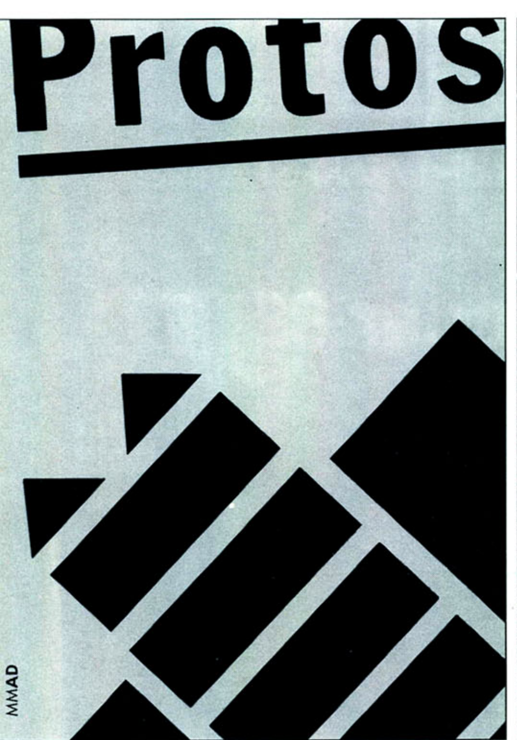
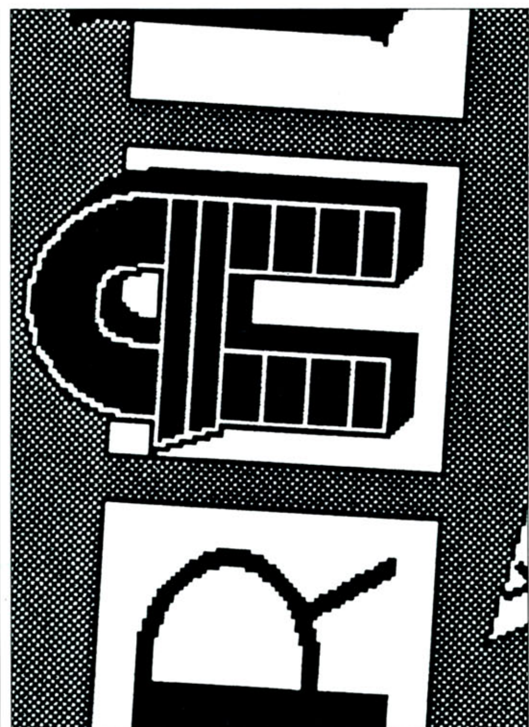

ATARI
ST COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
Januar 89
DM 7,-
Ös. 56,- Sfr. 7,-
1
CREATOR
Zeichnen & Animieren mit dem ST
NEODESK
Der alternative Desktop
ST-Vernetzung über MIDI
Listing
Der neue OMIKRON.Assembler







Das Bildschirm Survival-Set.

Protos ist der Name einer neuen Software aus der „Utility Series". *Protos* wird aus dem Autoordner gestartet und bleibt speicherresistent. *Protos* ist sozusagen das Schweizer Taschenmesser unter den Utilities, man hat viele Funktionen in einer Hand, die das Arbeiten am ST-Bildschirm angenehm komfortabel werden lassen. Man habe es also – wie das Taschenmesser – am besten immer dabei. Die Funktionen in aller Kürze: Mit der *Zoom*-Funktion läßt sich der Bildschirmausschnitt 2 – 8fach vergrößern oder verkleinern. Sehr gut für Übersichten oder Detailarbeiten. *Parameter* können *abgespeichert*, ein *Reset über die Tastatur* ausgelöst werden. *Protos* erlaubt weiter *die Definition von Makros,* die dann an Drucker, Midi oder an die serielle Schnittstelle geschickt werden können. Mausbewegungen oder Tastaturcodes speichert man einfach ab. Ein zuschaltbarer (und wieder abschaltbarer) Turboeffekt für die Maus: *Quick-Mouse.* Natürlich gibt es auch eine ein- und ausschaltbare Uhr und noch vieles andere mehr. N o c h mehr steht im Software-Info *Protos,* das wir auf Anfrage gerne zusenden.
Protos kostet 69,– DM.

Scarabus.
Die Rückkehr der Schriftbastler.

Scarabus ist ein Fonteditor. Man kann damit Signum!-Zeichensätze erfinden und vorhandene bearbeiten. Interessant ist dabei, daß *Scarabus* die Fontformate für alle Druckertypen (9-, 24-Nadel und Laserdrucker) in einem berücksichtigt und damit die drei Signum!-Fonteditoren ersetzt. Doch nicht genug damit: *Scarabus* nimmt sich nicht nur einzelner Zeichen an, was natürlich sehr schön ist, sondern ist auch in der Lage, Veränderungen am ganzen Zeichensatz auf einmal auszuführen: die Zeichensätze können konvertiert werden (aus einem Zeichensatz für 24-Nadler wird beispielsweise einer für Laser), können ihre Proportionalität verlieren, flugs schattiert, verdickt oder abgeschrägt werden. Durch das Mixen verschiedener Funktionen sind der Phantasie keine Grenzen gesetzt. Noch ein Clou: Teile von gescannten Bildern, Zeichen oder Signets können in einen Zeichensatz übernommen werden!
Scarabus ist mehr und kostet 100,– DM.

Neu:

Creator
Zeichnen, Konstruktion, Animation, Zeichentrickfilm: Grafik gekonnt in Bewegung gesetzt.

249,– DM.

Bolo spielt. Basteln da welche?

Bolo, das etwas andere Ballerspiel, ist eine Creation aus dem Hause Dr. Mausklick. Mittlerweile ist *Bolo* allseits beliebt und man muß immer noch mit Geschick über 50 Ebenen dorthin gelangen, wo der Mega-Ghost wohnt.

Für diejenigen, denen *Bolo* den Schweiß auf die Stirne getrieben hat, haben wir eine gute Nachricht und für alle Mausartisten, die *Bolo* mit links bewältigen (angeblich, sagt unser Hans-Holger), ebenso. Natürlich handelt es sich um dieselbe Nachricht:

Jetzt gibt es die *Bolo Werkstatt,* ein flottes Utility, mit dem die ersteren endlich über das verflixte dreizehnte Level hinauszukommen in der Lage sein werden. Den anderen ermöglicht es, sich selbst so viele Hürden aufzustellen, daß sie aus dem Mausstolpern nicht mehr herauskommen und sich der gewünschte Lustgewinn beim Spielen wieder einstellt.

Alle bedauernswerten Personen, die noch gar kein *Bolo* haben, haben die Möglichkeit, sich mit der *Bolo Werkstatt* eins zu basteln (zum Spielen benötigt man jedoch weiterhin ein Original-*Bolo*).

Bolo kostet 69,– DM, die *Bolo Werkstatt* gibt es überall da, wo es auch *Bolo* gibt und kostet ebenfalls 69,– DM.



Englerstraße 3
D-6900 Heidelberg
Telefon (0 62 21) 30 00 02
Telefax (0 62 21) 30 03 89

EDITORIAL

WO BLEIBEN DIE TRANSPUTER?

Mein Friseur hat aufgeschlagen. Doch das interessiert Sie sicher nur am Rande.
Aber es brennen uns auch noch andere Probleme unter den Nägeln[1], die wir leider an dieser Stelle **nicht** behandeln können.

So wollen wir uns hier und heute nicht schon wieder am klassischen Sprachstreit über das Edierproblem beteiligen, obschon ein vergrätzter Leser der Redaktion dräute, unser geliebtes Blatt nicht mehr zu kaufen oder gar zu lesen, wenn die Redaktion nicht endlich bereit wäre, das Wort *edieren* (lat. edere) falsch zu schreiben, was wir aber im Hinblick darauf, daß wir dann auch agitieren, wenn wir agieren meinen, sagen müßten, ablehnen.

Man hat Gewalt, so hat man Recht!
(Faust II, Goethe)

Und Sie haben immer noch keinen vollelektronischen Turnschuh (oder besser deren zwei) mit hinterleuchteter 4*25-Zeichen-Plasmaanzeige für Laufrhythmus, Pulsfrequenz, Körpertemperatur und Achselnässe sowie dreistimmigem Signalton beim Überschreiten Ihrer charakteristischen Fußschweißentwicklung, nicht zu vergessen den 5-Watt-Halogenkniestrahler für Nachtbetrieb? Läßt sich Ihre Kaffeemaschine *online* schalten? Sind Sie sicher?

Wir auch nicht, aber über solche Blüten würden wir wohl gerne in einem anderen Artikel lächeln. Lassen wir das also.

Prometheus brachte uns das Feuer,
Pandora den Aschenbecher.
(Unbekannter Dichter und Denker, um 1650)

Und wiewohl es uns über alle Maßen anficht, daß unser liebes Teutsch gar garstig Gewalt erfahren muß und wir uns nur noch kärglich am vertrauten Sprachschatz letzen können, nein, stattdessen von *Usern*[2] lesen müssen, die nichts anderes wissen, als *Windows* über *Screens* (und ihre *Computer* über den *Tisch*, Verzeihung: Tisch) zu ziehen: Wir ermangeln des Platzes und der Zeit, unser Darben ob der ungezählten Anglizismen in der einschlägigen (sic!) Fachliteratur zu schildern; im übrigen steht es uns gar nicht zu, solcherlei Schelte zu äußern - ertappen wir uns doch selbst hie und da beim lätscherten Sprachstyling.

Kühle Stühle in der Schwüle
fühlen sich kalt an.
(Unbekannt verzogen)

Aus diesem unseren Grunde rührt es uns auch nicht so sehr an, daß wir immer wieder "der Virus" lesen müssen (wie es der Duden inzwischen Millionen von Fliegen erlaubt), obwohl nach allem Menschenverstand das Neutrum die quälende Wahl für sich entscheiden muß.

Aber dieser Sprachvirus hat sich auch schon in diese Redaktion eingeschlichen, wenn auch seltener als anderswo.

Man lernt ständig aus.

Gegen Windkraftwerke zu Felde zu ziehen, stünde uns gleichwenig gut zu Gesicht: Immer wieder eräugnet es sich, daß Menschen jeder Hautfarbe, jeder Glaubenskongregation, jedweden Geschlechtes und jeder Schuhgröße zu uns kommen, in der Hoffnung, daß ihnen, die da kommen, Klarheit zuteil werde über die immerwährende confusio compatibile, wes Rechnerfabrikat zu wählen ihnen bestimmet sei. Meint man, sie hätten sodann ihre Meinung gegen die unsrige ausgetauscht, wenden sie sich sogleich hinter unser aller Rücken wider den Beschluß, den Paradiesvögeln der Ornis ein begehrend Auge zuzuwerfen, statt der Massentierhaltung Marke Big Flu den Zuschlag zu geben - und schon steht der Klon auf dem Tisch ("Ja, da kann man so schön Texte eingeben. Und kompabeatle bin ich dann auch noch."). Es bricht uns das Herz.

Chianti in flagranti, Martini im Bikini, Cola aus der Dose.

Allein, wie schon gesagt: Das soll uns heute alles nicht beschäftigen.

Nein, nein, nein, eigentlich geht es uns heute um etwas ganz anderes: *[wegen Überlänge gekürzt]*

Claus Brod

[1]Metapher!

[2]User, der: (engl.) jmd., der Drogen nimmt
(zitiert nach Duden 1986, 10. Auflage)

INHALT

SOFTWARE

HARDWARE

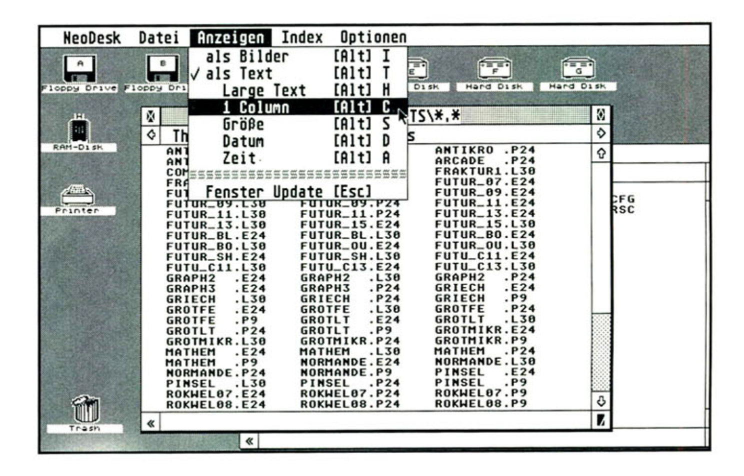
GRUNDLAGEN



The Creator

Zu bewundern war der Creator im Entwicklungsstadium schon auf diversen Messen. Nach fast zwei Jahren ist er nun endlich fertig, und man kann sagen, daß sich das Warten gelohnt hat. Aber für alle Nichteingeweihten sei schon mal erwähnt, daß der Creator ein Zeichenprogramm ganz besonderer Art für hohe Auflösung (monochrom) ist. Neben allen möglichen (und unmöglichen) Bearbeitungsmodi ist es mit ihm auch möglich, die erzeugten Grafiken in Bewegung zu setzen, denn der Creator verfügt über einen Zeichen- und einen Animationsteil. Doch alles der Reihe nach.

Seite **28**



NeoDesk
- Das alternative Desktop

Vor einigen Monaten konnte man in den amerikanischen Mailboxen über ein neues Produkt namens NeoDesk lesen. Die Anwender überschlugen sich geradezu mit Lobhuldigungen zu diesem Programm. Auf der Düsseldorfer ATARI-Messe war es zum ersten Mal in einer Vorabversion in Deutschland zu sehen. Jetzt ist es ganz offiziell erhältlich.

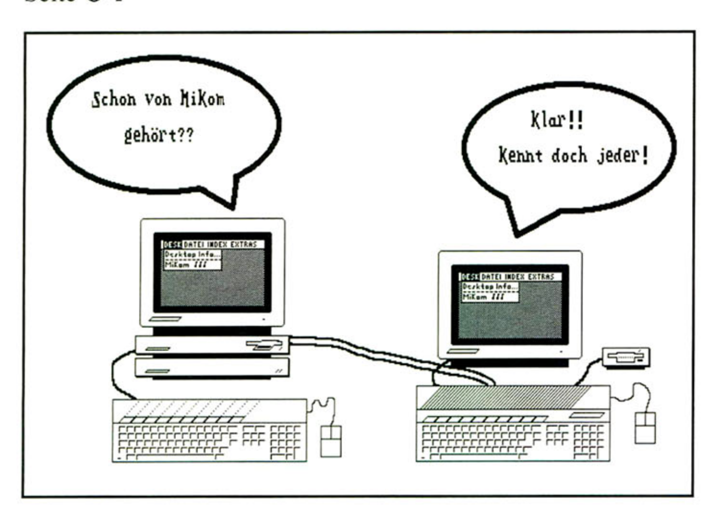
Seite **154**



Schon wieder Malprogramme

Mal- und Zeichenprogramme auf dem ATARI ST gibt es fast schon wie Sand am Meer - und doch kommen immer wieder neue auf den Markt! Was war der Freak zu Beginn des ATARI ST-Zeitalters doch froh, wenn er wenigstens ein paar Linien mühsam mit dem mitgelieferten ST-BASIC auf den Bildschirm brachte. Zu jener Zeit war ein komfortables Malprogramm noch Mangelware. Dies erkannten die Softwarehäuser und reagierten natürlich binnen kürzester Zeit darauf. Nach und nach erschienen immer mehr Grafikpakete auf dem Markt, und so fielen auch die Preise auf einen erträglichen Level. Inzwischen sind gute Zeichenprogramme für unter 100,- DM erhältlich, die (fast) alle wünschenswerten Funktionen in sich vereinen.

Seite **64**



Zwei Computer gekoppelt

Jetzt, wo die Megas so richtig in die heimischen Computerstuben Einzug gehalten haben, finden sich sicherlich bei vielen ST-Freunden zwei STs. Aber nicht nur Besitzer zweier STs sollen angesprochen werden, sondern auch alle diejenigen, die mal zwei Rechner miteinander verbinden möchten.

Seite **102**

PROJEKT

PROGRAMMIERPRAXIS

ANWENDUNGEN

AKTUELLES

RUBRIKEN

NEWS

Kyrillisch auf dem ST

Ab sofort ist für alle Russisch-Freunde ein Programm zur Darstellung kyrillischer Zeichen auf dem ATARI ST und jedem EPSON-kompatiblen Nadeldrucker erhältlich. Mit dem *Saporoschje ST*-Accessory lassen sich in fast jedem beliebigem Programm (z.B. 1st_Word, DBase, Adimens, GFA BASIC, STAD, Tempus etc.) über Tastendruck wählbar kyrillische oder lateinische Buchstaben gemischt eingeben und sofort richtig auf dem Bildschirm darstellen.

Fast alle nationalen Sonderzeichen des ATARI-Systemzeichensatzes lassen sich nun auch über Tastatur aufrufen. So kann man für fremdsprachige oder gemischte Texte weiterhin seine Lieblingstextverarbeitung benutzen. Allerdings ist Saporoschje auch für Vokabeltrainer, Datenbanken etc. ganz interessant.

Die Ausgabe der Zeichen auf einem Drucker erfolgt maximal in einer Auflösung von 240 x 216 DPI, Proportionalschrift ist ebenfalls möglich. Trotz der Zeichenausgabe im Grafik-Modus werden auch Textattribute (Schmal, Fett,...) korrekt dargestellt.

Geliefert wird Saporoschje ST mit fünf NLQ-Zeichensätzen, einer ausführlichen Dokumentation, einem Zeichensatzeditor und einem Zeichenkonverter für DM 249,-, eine 'student version' mit nur einem Zeichensatz, ohne Editor und Konverter ist für DM 149,- erhältlich.

CTS Haustein
Teutoburger Str. 93
4200 Obernhausen 11



16mal logisch

Einen preisgünstigen Einstieg in die Logikanalyse bietet ein neues Softwareprodukt für alle ATARI STs. Schon mit 512kB Speicher können bis zu 16 Kanäle gleichzeitig eingelesen und gespeichert werden. Die dabei entstehenden Kurven können miteinander verglichen, logisch verknüpft und ausgedruckt werden.

Der hochauflösende S/W-Monitor erlaubt die Darstellung von insgesamt 160 Kanälen. Durch die Beschriftungsmöglichkeit jedes einzelnen Kanals mit Text ist das Messergebnis trotzdem übersichtlich und aufschlußreich. Die 330KHz Echtzeitabtastung des Programms kann durch einen Hardwarevorsatz auf 40 MHz erweitert werden. Damit wird das Gerät dann auch allen professionellen Ansprüchen genügen. Als Ein- und Ausgabekarte wird lediglich das weitverbreitete Userport-Interface aus dem Heise-Verlag verwendet, jedoch läßt sich das Programm auch mit allen anderen Ein-/Ausgabekarten betreiben

TDM-Elektronik
Luitpoldplatz 23
8580 Bayreuth
Tel.: 0921/82590



MIDI-Recording

Mit 1st Track *bietet die Berliner Firma Geerdes eine neue MIDI-24-Spur-Recording Software für alle ATARI ST mit monochromem Monitor. Sie kann bis zu einer 1536-tel Note auflösen. Fehlerhafte Aufnahmen können per Locatoren, die auch abschaltbar sind, auf den Punkt genau überspielt werden. Für den Playbetrieb steht ein Cycle-Modus für endlose Wiederholungen zur Verfügung. Zur Zeitkorrektur ungenauer Einspielungen stehen die üblichen Quantisierungsstufen zur Verfügung. Eine erweiterte Version 2nd Track ist bereits in Arbeit. Wer an 1st Track interessiert ist, kann es für DM 128,- unter untern stehender Adresse erwerben. Eine Demoversion ist gegen DM 5,- erhältlich.*

G.C.Geerdes
Bismarckstr. 84
1000 Berlin 12
Tel.: 030/316779



voilà 2.0

So heißt eine leistungsfähige, flexible Datenbank der Firma Maxisoft. Das Programm ist speziell zur Erstellung zur Erstellung umfangreicher Literatur-, Material-, Text- und Informationsdateien ausgelegt. voilà unterscheidet sich von anderen Datenbanken insbesondere in der Möglichkeit, zu jedem Datensatz eine Schlagwortkarteikarte erstellen zu können.

Es stehen dem Anwender sechs normale Eingabe-, fünf Schlüsselfelder, ein Datum- und ein Nummern- oder Zahlenfeld zur Verfügung. Zu jedem Datensatz können maximal 10 Schlagwörter abgespeichert werden. voilà ist zum Preis von DM 99,- bei folgender Adresse zu erhalten:

Maxisoft
F.Schuhmann
Feldstr. 27
3078 Stolzenau
Tel.: 05761/3772



Soundmachine ST

Tommy Software hat den offiziellen Nachfolger von MusiX32, die *Soundmachine ST*, auf den Markt gebracht. Es sind mehrere Instrumente (mit sehr gutem Klang) gleichzeitig spielbar. Die Eingabe erfolgt ähnlich wie bei MusiX32 einfach per Maus. Ausgeben kann man die erzeugte Klänge auf Monitor oder Stereoanlage. Ferner ist ein Drumcomputer integriert. Mitgeliefert werden Beispiele zur Einbindung von Musikstücken als Hintergrundmusik in mehreren gängigen Programmiersprachen.

Die Soundmachine ST läuft auf Farb- und Monochrommonitoren und allen ATARI ST-Rechnern mit beliebigem Betriebssystem. Der Verkaufspreis beträgt DM 148,-.

Zusätzlich ist eine Sounddiskette SoundLib01 mit einer Vielzahl zusätzlicher Instrumente und Schlagzeug-Kits für DM 79,95 lieferbar.

Tommy Software
Selchower Str. 44
1000 Berlin 44
Tel.: 030/6214063



eLAN

The Industrial NET für ATARI ST

Nachdem bereits auf der CeBIT' 88 in Hannover und auf der ATARI-Show in Düsseldorf erste lauffähige Vorversionen des Netzwerkes eLAN - The Industrial NET der GTI GmbH, Berlin, für ATARI ST-Computer vorgestellt wurden, hat dieses Produkt nunmehr Marktreife erlangt.

Das eLAN-Netzwerk, das ursprünglich für den Einsatz im industriellen Umfeld konzipiert wurde, ist das erste lokale Netz für ATARI ST-Computer, das konsequent unter Einhaltung internationaler Normen realisiert wurde. So folgt eLAN den Empfehlungen der MAP-Spezifikation für echtzeitfähige Industrienetze und basiert auf dem internationalen Protokoll-Standard PROWAY C.eLAN ist heuer mit dem "Berliner Innovationspreis 1988" ausgezeichnet worden.

Das eLAN -Netz arbeitet nach dem Token-Passing-Prinzip mit einer Übertragungsrate von 1 MBit/s. Es können maximal 254 Stationen über eine Entfernung von 2000 m angeschlossen werden. Aufgrund seiner hohen Störsicherheit verträgt sich eLAN mit verschiedenen Übertragungsmedien, vom Twisted-Pair- über Koaxialkabel bis hin zur LWL-Verbindung. Größere Entfernungen können durch Repeater-Stationen erreicht werden.

Das Netzwerkbetriebssystem auf dem ATARI ST wurde vollständig in C implementiert. Es beinhaltet die komplette normgerechte PROWAY-C-Schnittstelle und verhält sich gegenüber Applikationsprogrammen 100% kompatibel zum TOS. Es stellt alle notwendigen Netzfunktionen zur Verfügung und beinhaltet selbstverständlich File- und Record-Lock-Mechanismen. Mit eLAN vernetzte STs können in gewohnter Art und Weise über das virtuelle Laufwerk N: angesprochen werden. Alle bekannten Desktop-Funktionen laufen auch über das Netzwerk. Wegen der konsequenten Einhaltung der von ATARI freigegebenen Systemschnittstellen, arbeiten auch alle existierenden Applikationsprogramme, die sich ebenfalls an diese Konventionen halten, sofort im Rahmen ihrer Möglichkeiten mit eLAN zusammen.

Für spezielle mehrplatzfähige Anwendungen stellt das Netzwerkbetriebssystem dem Programmierer umfangreiche Funktionen zur Verfügung. Einige bekannte Software- und Systemhäuser haben sich bereits für den Einsatz von eLAN entschieden (z.B. Bavaria Soft Datentechnik mit BSS-Plus, Biosystems SRI GmbH mit MEDICOM usw.).

Obwohl hier für ATARI ST-Computer vorgestellt, ist eLAN ein herstellerunabhängiges Kommunikationssystem. eLAN-Netzknoten sind derzeit für ATARI ST, für IBM PC/AT, für VMEbus- und ECB-Systeme sowie universell als V24-Koppler in der Produktion. Weitere Rechnersysteme, wie Apple Macintosh und DEC VAX, werden in KÜrze folgen. Durch die konsequente Einhaltung von Normen und Standards ist eine problemlose Portierbarkeit auf die verschiedenen Betriebssysteme gewährt. eLAN stellt somit eine sichere Investition auch für die Zukunft dar.

Bis Mitte 1989 wird es bereits möglich sein, ATARI STs zusammen mit MS-DOS-PCs in einem gemeinsamen eLAN-Netzwerk zu betreiben. Die Auslieferung für ATARI ST beginnt aber bereits im Dezember 1988.

GTI
Gesellschaft für technische Informatik GmbH
Unter den Eichen 108a
1000 Berlin 45
Tel.: 030/8315021/22



Ergänzung zum Tastaturenvergleich *aus Heft 12/88*

Die Firma Ruff & Locher wird im Januar 1989 einen PC ditto-Patch fertiggestellt haben, womit der PC-Emulator auch mit ST-TAST zu bedienen ist. Ab Erscheinen dieser Ausgabe soll er im Lieferumfang enthalten sein.

Im Test wurde das Programm EDITTAST nicht erwähnt. Es erlaubt die individuelle Belegung der PC-Tasten mit bis zu vier Zeichen.

Das Phänomen des Slashs auf der Bindestrichtaste müßte, nach Aussage von Ruff & Locher, durch ein versehentliches Betätigen der NUM-Lock-Taste ausgelöst worden sein. Dabei werden, mit der mitgelieferten Tastaturbelegung, nicht nur den Tasten des Zehnerblocks neue Codes zugeordnet, sondern auch der Bindestrichtaste. Diese Einstellung kann mit EDITTAST.PRG geändert werden.

Ein zweiteiliges Steckgehäuse ist erhältlich. ST-TAST arbeitet auch mit einem ROM-Port-Expander zusammen, wodurch der ATART ST gleichzeitig andere Erweiterungen ansprechen kann.

Ruff & Locher Datentechnik
Eichachstr. 13
7404 Ofterdingen
Tel.: 07473/22810



ADIMENS und ADITALK

ADIMENS ST 2.3 sowie ADITALK ST 2.3 werden jetzt zusätzlich zum ATARI-Systemfachhandel von ADI direkt vertrieben. Der Preis beträgt jeweils DM 249,-.

Bisherige ADIMENS- und ADITALK-Kunden können durch Einsendung des Originalpakets an ATARI für DM 99,- die neuen Versionen als Update erhalten.

ADI Software GmbH
Bunsenstr. 22
7500 Karlsruhe
Tel.: 0721/82030

Graphik Artist für 19"-Monitor

Das CAD-Programm Graphik Artist ist jetzt auch ohne Aufpreis für 19"-Monitore erhältlich. Besondere Features sind:

- *256 Layer*
- *Zoombereich von 1:10000000*
- *Rotieren um jeden Winkelgrad*
- *Spreadsheet und Graphikauswertung*
- *Makroverarbeitung uvm.*

Für DM 20,- ist eine Demoversion erhältlich. Der Verkaufspreis beträgt DM 698,-.

KFC Computer
Wiesenstr. 18
6240 Königstein
Tel.: 06174/3033



aladin 3.0

Für das alternative Betriebssystem *aladin*, das den Apple Macintosh emuliert, kommt dieser Tage eine neue Version 3.0 auf den Markt. Sie unterstützt nun endlich eine Festplatte, wobei Original-GEM-Partitionen benützt werden, so daß man sowohl normale ATARI- als auch Macintosh-Programme zusammen auf einer Partition verwenden kann. Es wird der volle ACSI-Standard erfüllt, d.h. bis zu acht Kontroller mit bis zu acht Festplatten werden unterstützt. Die Partitionsgröße kann unter aladin bis zu 512 MB betragen. Ein neuer Chip auf der aladin-Platine regelt dabei den schnellen Festplattenzugriff.

Ebenfalls wird die komplette Macintosh-Soundhardware emuliert. Der Sound wird wahlweise über den Druckerport des ST oder den ATARI-Monitor ausgegeben. Benutzt man den Druckerport, kann ein Ferner wird jetzt auch das Macintosh-System 4.1 unterstützt. Da manche ATARI-Laufwerke einen Diskettenwechsel nicht mehr automatisch erkennen, ist die Diskettenverwaltung aladins geändert worden; man kann jetzt zwischen automatischer und manueller Verwaltung wählen. Erweitert wurden auch die Formatierungsmöglichkeiten von aladin; es sind jetzt 400 kB, 800 kB und Janus Format-Disketten möglich. Bei letzterem kann man auf der Vorderseite der Diskette ATARI-GEM- und auf der Rückseite aladin-Format verwenden.

Neu ist auch die Möglichkeit, Subdirectories anzulegen und den Blitter des ST zu nutzen. Zuletzt sind eine ganze Reihe neuer Druckertreiber hinzugekommen. So werden jetzt die NEC P6, P5 usw., der Epson FX 80 & Kompatible, der ATARI Laserdrucker und HP Laserjet und Kompatible unterstützt. Dazu werden Bildschirm- und Druckerfonts mitgeliefert.

Das Update von Version 2.1 auf Version 3.0 kostet DM 60,-. Für Besitzer der Version 1.3 kostet das Update DM 130,-. Eingeschickt werden muß die Original-aladin-ST-Diskette, ein Scheck über den jeweiligen Betrag und ein ausreichend frankierter Rückumschlag. Zusätzlich sollte man angeben, welchen Druckertreiber man benötigt, da er nur für einen Drucker kostenlos mitgeliefert wird.

Softpaquet
Weteringdreef 61
NL-2724 GT Zoetermeer
Tel.: 0031/79/423571



D-50 Sorcerer...

ist ein extrem leistungsstarker Editor für die Roland Synthesizer D-50 und D-550, lauffähig auf allen ATARI ST ab 1 Megabyte RAM. Er läuft in Farbe und monochromer Auflösung. Mit ihm lassen sich mittels vier Algorithmen kinderleicht Sounds generieren, die sich natürlich später auch verwalten und edieren lassen. Der D-50 Sorcerer ist einfach zu bedienen und liegt mit DM 150,- im Low-Cost-Bereich.

Peter Linsener
Softwareentwicklung
Werrastr. 42
1000 Berlin 44
Tel.: 030/6877220



PC ditto 3.96
MS-DOS-Emulator

Neue Version 3.96
In Europa nur
exklusiv von MAXON



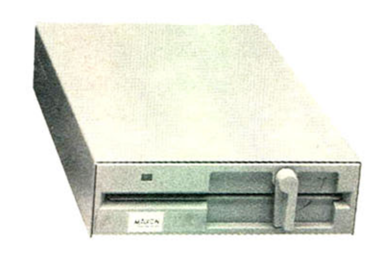
Schwarz auf Weiß kann jetzt jeder auf seinem ATARI ST unter MS-DOS arbeiten. Doch nicht nur monochrom, auch in Farbe ist nun der Zugriff auf die Welt der PC-Rechner möglich.

Die Software-Emulation **PC ditto** öffnet allen ATARI ST-Anwendern das Tor zum gewohnten professionellen Business Standard.

Mit dem **PC ditto** können Sie mühelos mit Lotus 1-2-3 oder Symphony Ihre Kalkulation erstellen oder Ihre Daten mit DBase III plus verwalten. Mit dem **PC ditto** haben Sie Zugang zu Turbo Pascal, zum GW BASIC Interpreter, und auch zu Borlands neuestem Kind Turbo BASIC.

Mit dem **PC ditto** laufen so viele Programme, daß wir sie hier gar nicht alle auflisten können. Fordern Sie eine Liste an.

Der **PC ditto** unterstützt nahezu alle Festplatten, die an den ATARI ST anschließbar sind, den Druckerport und alle Schnittstellen, soweit es die Hardware zuläßt. Machen Sie aus Ihrem ATARI ST den preiswertesten PC-CLONE!



Passend zu **PC ditto** bieten wir ein 5 $^{1}/_{4}$"-Markenlaufwerk der Firma NEC für DM 398.- an. Es hat 40/80 Spuren-Umschaltung und wird anschlußfertig für den ATARI ST geliefert.

Mit 48- seitigem, deutschem Handbuch!

Update auf Version 3.96

In der neuen Version 3.96 wurden u.a die Tastaturunterstützung und der Festplattenbetrieb mit ATARI-fremden Geräten verbessert. Natürlich wurde auch dafür gesorgt, daß noch mehr Programme der MS-DOS-Welt mit PC ditto problemlos laufen.

Bei der MAXON Computer GmbH registrierte Kunden haben die Möglichkeit ein Update auf die neue Version 3.96 gegen eine Bearbeitungsgebühr von DM 10,- zu erhalten. Dazu ist die Einsendung des Betrages als Scheck, der Originaldiskette und eines ausreichend frankierten und bereits adressierten Rückumschlags notwendig.

Sollten Sie über eine ältere PC ditto-Version verfügen, der **keine Registrierkarte der MAXON oder Merlin Computer GmbH beilag**, bieten wir einmalig bis zum 31.01.1989 den Umtausch gegen die neue Version für DM 49,- an. Zum Umtausch senden Sie uns **Diskette und Verpackung** der älteren Version sowie einen Scheck über DM 49,-. Sie erhalten dann eine neue Diskette, das Handbuch und eine MAXON-Registrierungskarte.



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ____________

Vorname: ____________

Straße: ____________

Ort: ____________

Unterschrift: ____________

Hiermit bestelle ich:

☐ PC ditto wie oben beschrieben für DM 198.00

☐ 5 $^{1}/_{4}$"-Laufwerk für ATARI ST für DM 398.00

Versandkosten: DM 7,50

Nachnahme zzgl. DM 3,50 Nachnahmegebühr

☐ Vorauskasse

☐ Nachnahme

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrieb in Holland: JOTKA COMPUTING Postbus 8183 NL-6710 AD Ede Tel.: 08380/38731 Fax.: 08380/21675

Turbo ST Der Softwareblitter

Was viele nicht für möglich hielten, ist jetzt verwirklicht. Die schnellere Textausgabe, die durch den Blitterchip im ATARI ST bewirkt wird, kann mit *Turbo ST* durch Software simuliert werden. Das Besondere dabei ist, daß Turbo ST teilweise sogar noch schneller als der Blitter arbeitet. So können Beschleunigungen von über 650% gegenüber der normalen Textausgabe ohne Blitter erreicht werden.

Turbo ST wird als Accessory geladen und kann wie der Blitterchip abgeschaltet werden. Gerade beim Programmieren oder Schreiben in Textverarbeitungen, die die normalen Routinen des ST benutzen, lassen sich so kürzere Darstellungszeiten erzielen.

Besonders interessant ist Turbo ST für Besitzer der älteren Rechnertypen wie 260ST, 520ST und 1040ST, in denen die Nachrüstung nicht vorgesehen oder mit großen Kosten verbunden ist. Kein Löten und kein Garantieverlust. Das Programm und Handbuch sind in Deutsch. Der Preis beträgt *DM 79,-*.

Ebenfalls wurde das Terminalprogramm *INTERLINK ST* verbessert. In der nun vorliegenden Version 1.89 konnten schon viele der durch deutsche Benutzer gemachten Verbesserungsvorschläge eingearbeitet werden. Die in Version 1.85 noch vorhandenen Fehler wurden beseitigt. So sind z.B. die Probleme mit den deutschen Umlauten nicht mehr vorhanden. Registrierte Benutzer sind in der Zwischenzeit schon benachrichtigt worden.

Auf vielfachen Wunsch wird z.Zt. an dem Übertragungsprotokoll ZMODEM gearbeitet, das wahrscheinlich im nächsten Upgrade schon enthalten sein wird. Als besonderes Bonbon wird dann ein Editor mitgeliefert, mit dem der Anwender Terminalemulationen und Übertragungsprotokolle selber schreiben kann. Der Preis beträgt nach wie vor *DM 79,-*.

Von der gleichen Firma kommt ein neues Utilityprogramm *Revolver* für *DM 129,-* auf den Markt, das unter anderem die Partitionierung des Speichers (einen "Switcher") und das Abspeichern einer Partition auf Diskette oder Festplatte ("Roll Out" und "Roll In") erlaubt. Damit ist z.B. das Umschalten vom Grafikprogramm in die Tabellenkalkulation in 2 Sekunden möglich. Ein ST kann in bis zu 8 Partitionen aufgeteilt werden. Es ist das erste Programm, welches über resetfeste Partitionen verfügt; d.h. ein Reset betrifft nur die gerade benutzte Partition und läßt die anderen unberührt!

Dazu kommen natürlich nützliche Hilfen wie resetfeste RAM-Disk, resetfester Druckerspooler (der löschbar ist!), Diskutilities, Warm- und Kaltstart und erweiterte Kontrollfeldfunktionen. Darunter sind solche kleinen aber feinen Dinge wie der (natürlich abschaltbare) simulierte Doppelklick durch einfaches Drücken der rechten Maustaste.

Ein "Roll Out" speichert den kompletten Inhalt einer Partition mit allen darin enthaltenen Programmen - also auch Hintergrundprogrammen und Deskaccessories - mit allen zugehörigen Systemparametern auf Diskette, Festplatte oder RAM-Disk für eine spätere Weiterbenutzung ab. Die Abspeicherung selbst erfolgt komprimiert. So liegt die Größe einer 1 MByte-Partition typischerweise bei ca. 350 kByte. Wird eine solche Datei über "Roll In" wieder geladen, ist eine Weiterarbeit an der gleichen Stelle möglich. Die Ladezeit liegt bei ca. 20 Sekunden (Festplatte).

BELA Computer GmbH
Unterortstr. 23-25
6236 Eschborn
Tel.: 06196/481944



12 MB-Laufwerke auch für den ST

Die Verbatim Corp. hat vor rund einem Jahr erstmals ein 12 MB-Laufwerk als primären Massenspeicher oder leistungsfähiges Backup-System vorgestellt. Viele Anwender von ATARI ST-Rechnern suchten seitdem nach Möglichkeiten, dieses Laufwerk ebenfalls zu nutzen. Aus diesem Grund hat Verbatim in Zusammenarbeit mit 3K-EDV-Entwicklungen einen ATARI-Host-Adapter entwickelt, der eine Verbindung mit diesen Geräten erlaubt.

Als Speichermedium wird in diesem Laufwerk eine 5 1/4"-Diskette verwendet, die speziell durch ihr Hardcover einer 3,5"-Diskette ähnelt. Sie ist werkseitig unter TOS vorformatiert und bietet so 9,54 MB formatierte Speicherkapazität. Durch einen DMA-Ausgang können zusätzliche Geräte und Kontroller angeschlossen werden.

Das Laufwerk kann auch als Bootlaufwerk anstelle einer Festplatte verwendet werden. Es ist datenkompatibel zum universellen Massenspeichersystem von 3K und DOS 3.3. Ein Medienwechsel vom Desktop und aus laufenden Programmen ist ohne Gefahr eines Datenverlustes durchführbar.

Das mitgelieferte Backup-Utility erlaubt extrem schnelle Backups. Die Übertragungsrate beträgt 350 KB/s. Eine mittlere Zugriffszeit von 65 ms kommt einer Festplatte gleich.

Verbatim GmbH
Frankfurter Str. 63-69
6236 Eschborn



Festplattenpalette bei Eickmann erweitert

Ab sofort sind bei der Frankfurter Firma Eickmann Computer Festplatten mit einer Speicherkapazität bis zu 110 MB erhältlich. Alle Geräte werden mit einem Interleavefaktor von 1 formatiert, die eine maximale Geschwindigkeit (größer 750 kB/s) der Datenübertragung garantieren. Sie sind so aufgebaut, daß sie eine möglichst niedrige Geräuschentwicklung haben (ca. 43 dB(A)) und werden mit 74-seitigem deutschem Handbuch und komfortabler Treibersoftware geliefert. Besonders erwähnenswert sind die Möglichkeiten des Treibers, einzelne Partitionen mit einem Schreibschutz zu versehen und von verschiedenen Partitionen zu booten.

Die Festplatten können zu folgenden Preisen im Fachhandel oder direkt bei Eickmann Computer erworben werden:

30 MB	DM 1498,-
60 MB	DM 2098,-
80 MB	DM 2998,-
110 MB	DM 3490,-

Eickmann Computer
In der Römerstadt 249
6000 Frankfurt 90
Tel.: 069/763409



Mausmakros, Lupe und großer Bildchirm; Protos.

Protos. ist ein neues Programm in der Utility-Serie von Application Systems Heidelberg. Mit *Protos.* kann jederzeit eine Lupe mit drei verschiedenen Vergrößerungsstufen aufgerufen werden. Dadurch erhalten alle Programme eine sehr schnelle Lupenfunktion, und pixelgenaues Arbeiten wird in jedem Programm möglich. Die Lupe selbst läuft in einem eigenen Fenster ab (kein GEM-Fenster). Zusätzlich kann jederzeit ein Optionsmenü aufgerufen werden, in dem man Makros für die Tastatur und die Maus eingeben kann. Mit Mausmakros können beliebige Folgen von Mausaktionen abgespeichert und beliebig wieder aufgerufen werden. Dadurch wird z.B. das Laden eines Zeichensatzes in Signum!2 mit einem Tastendruck möglich. Die Texte der Makros können auch an die serielle Schnittstelle oder über MIDI gesendet werden, so daß z.B. per Tastendruck ein Formfeed beim Drucker ausgelöst werden kann. Für Benutzer von DTP-Programmen ist die Simulation eines Großbildschirms interessant (bis 9984x9984 Pixel, ab Blitter-TOS), der im 640x400 ATARI-Bildschirm dargestellt wird. Zusätzliche Funktionen wie Zeit- und Datumsanzeige, Reset per Tastatur, Quickmaus, Bildschirmschoner und Sonderzeichen über zusammengesetzte Zeichen sind auch enthalten. Protos. kostet DM 69,-.

Application Systems Heidelberg
Englerstr. 3
6900 Heidelberg
Tel.: 06221/300002



Querdruck auf dem ATARI ST

Mit dem Programmpaket *Querdruck* können Texte um 90° gedreht auf 9- bzw. 24-Nadeldruckern ausgegeben werden. Dadurch läßt sich in vielen Fällen ein Zusammenkleben breiter Texte vermeiden. Vor allem in Zusammenarbeit mit Tabellenkalkulations-, Finanz- und Datenbankprogrammen ergeben sich Vorteile. Viel Wert wurde auf Schnelligkeit, klare Schriftzeichen und gute Bedienbarkeit gelegt.

Es lassen sich beliebig breite (max. 32767 Zeichen) und beliebig schmale Texte ausdrucken. Geeignet ist das Programmpaketsowohl für DIN A4- und DIN A3-Drucker. Es erfolgt eine Formatierung von Höhe und Breite. Kopfzeilen lassen sich (z.B. für Tabellen) wiederholen. Es können alle ATARI-Zeichen wiedergegeben werden, und man hat mehrere Schriftgrößen zur Verfügung.

Querdruck ist nicht kopiergeschützt und läuft auch auf Mega STs mit Blitter. Erhältlich ist es bei Philgerma-München-Dortmund. Der Autor bietet es auch im Direktvertrieb für DM 48,- an.

Entwicklungsbüro Dr.Ackermann
Kanalweg 1a
8048 Haimhausen
Tel.: 08133/1053



Universelles Massenspeichersystem

3K EDV-Entwicklungen stellte kürzlich das *U.M.S.S.*, das Universelle MassenSpeicher-System, vor. Herzstück des Systems ist ein von 3K entwickelter SCSI-Kontroller, der sämtliche Spezifikationen dieses Standards erfüllt. Dadurch ist jegliche Hardware mit SCSI-Interface anschließbar. Der Kontroller wird am DMA-Port des ST angeschlossen, natürlich ist auch ein DMA-Ausgang zum Anschluß weiterer Geräte vorhanden. Die ebenfalls von 3K entwickelte Treibersoftware unterstützt zwei dieser Kontroller, wobei an jeden bis zu acht Geräte angeschlossen werden können.

Wahlweise ist das System mit einem 44 MB Wechselplattenlaufwerk der Firma Syquest oder dem 10 MB Diskettensubsystem von Verbatim erhältlich. Optional wird das System mit einer zusätzlichen 20 oder 40 MB Festplatte ausgerüstet. Auf Wunsch werden Industriefestplatten beliebiger Größe bis 700 MB eingebaut. Um derartige Datenkapazitäten verwalten zu können, wurde ein neues Treiberkonzept realisiert, mit dem Massenspeicher mit bis zu 8 Gigabyte verwaltet werden können. Auch der Medienwechsel sowie der bei den Wechselmedien mögliche Schreibschutz wird von der Treibersoftware verwaltet. Darüber wurde die Möglichkeit geschaffen, Medien so zu formatieren, daß sie sowohl unter TOS auf dem ST als auch unter MS-DOS 3.3 auf PCs benutzt werden können, wodurch ein schneller Datenaustausch möglich wird. Weiterhin können Auto-Ordner und Accessories vom Massenspeicher gebootet werden. Der Treiber arbeitet auch neben anderen Massenspeichertreibern einwandfrei. Zusätzlich ist ein Backup-Programm im Lieferumfang enthalten, mit dem ganze Partitionen einer Festplatte schnell kopiert werden können.

3K EDV-Entwicklungen
Hülser Str. 76
4154 Tönisvorst 1
Tel.: 02151/700522



Hänisch Modula-2

Die Kasseler Firma Schwab Software vertreibt ein neues Modula-2 für den ATARI ST. Hänisch Modula-2 unterstützt z.Zt. die Prozessoren 68000, 60020 und 68881, in Vorbereitung sind 68030 und 68882. Es gibt Versionen für verschiedene Betriebssysteme auf dem ST:

TOS- Version	*DM 199,*
GEM- Version	*DM 298,*
RTOS-Version	*DM 445,-*
OS-9- Version	*DM 598,-*
(o.Editor	*DM 398,-)*

Schwab Software
Mühlbachweg 6
3500 Kassel
Tel.: 0561/402338



PICTURE-DISKS GRAFIKSAMMLUNG

Die Picture-Disks-Serie ist die größte Grafiksammlung die es für den Atari ST Computer gibt. Sie besteht aus hochauflösenden Grafiken für den Monochrombetrieb der Atari-Computer. Picture-Disks sind von der Qualität einzigartig und decken alle Themenbereiche ab.

1000 Graphiken DM 98, –
2400 Graphiken DM 198, –

Die Grafiken können direkt in das Programm STAD oder SIGNUM II eingelesen werden. Für andere Programme steht ein Accessory zur Verfügung, das die Grafiken in DEGAS- oder DOODLE-Format übersetzt. Damit steht für jeden Anwendungsbereich das passende Format zur Verfügung. Den Picture-Disks liegt jeweils ein Handbuch bei, das Ihnen ein schnelles Auffinden und Arbeiten mit den Picture-Disks ermöglicht.

Picture-Disks sind ein Muß für alle professionellen DTP-, Mal- und Textverarbeitungssysteme mit Grafikeinbindung!!!

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

Picture-Disks

Picture-Disks
Grafiksammlung

Erhalten Sie
im guten Fachhandel unverbindlich empfohlener Verkaufspreis
oder direkt beim Heim-Verlag

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

Ich bestelle
☐ 1000 St. **PICTURE-DISKS** zum Preis von **98, – DM**
☐ 2400 St. **PICTURE-DISKS** zum Preis von **198, – DM**

zzgl. DM 5,- Versandkosten (unabhängig von besteller Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________
Straße, Hausnr. ____________________
PLZ, Ort ____________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Statistik mit WiSTat 2.0

noch komfortabler – noch vielseitiger
alle einfachen Testverfahren (t-Tests usw.)
Varianzanalysen Korrelationen Regressionen
Faktor-, Cluster-, Item- Diskriminanzanalyse
Bequemes Dateienhandling Transformationen
mit Handbuch 398 DM
Studentensonderpreis 199 DM

Neu: WiSTat Graph
Der **Graphikeditor** für WiSTat 2.0 **99 DM**
Wenn Sie ein besseres Statistikpaket finden, dann kaufen Sie es!
fordern Sie unser **12 seitiges Info** an!

Graphik mit Prograph

Tabellenkalkulation und Graphik
komfortable Dateneingabe und Editierung
publikationsreife Druckqualität
Einlesen v. Fremdformaten (VIP, ASCII usw.)
Torten-, Balken-, Kurvendiagramme 2D 3D
automatische Achsenskalierung
Beschriftungen aller Diagramme, Legenden
Vergrößerung, Verkleinerung u.v.m.
DAS Chartprogramm für Ihren ST
mit Handbuch 248 DM · Demodiskette 10 DM

Informationen zu allen Programmen!
Händleranfragen erwünscht!

Texttuning für 1st Word/+ mit 1st TOOLS

Alle Befehle über Tastatur: **WORDCTRL.ACC**
Register u. Inhaltsverzeichnis: **1st INDEX.PRG**
Fußnoten ans Textende: **2nd FNOTE.PRG**
Fußnoten jede Seite neu: **PAGENOTE.PRG**
IBM Graphikzeichensatz: IBMFONT.PRG
Fremdtexte in 1st Word laden: **ASCIIWP.PRG**
6 Superutilities mit Handbuch 99 DM

REGRESSION ST 3.0
seit 2 J. an Universit. u. Klinik. im Einsatz
Komfortabler Editor, umfangreiche Graphiken
8 versch. Regressionen, Spline u.v.m.
mit Handbuch 198 DM · Demodiskette 10 DM

TEL SOFT EDV Entwicklung & Computersysteme
Thomas Leschner 3550 Marburg Universitätsstraße 40

Gigatron GbR
Resthauser Str. 128 · 4590 Cloppenburg
Tel. 04471-3070

2 – 4 Megabyte **intern**

ATARI 1040 ST

Speichererweiterung 2 Megabyte

Aufrüstbar auf 4 Megabyte

Aufgebaut mit 1 Megabit-Chips

2 Megabyte **998, – DM** ab Lager

Das hat Ihrem
CASIO® FX-850P
gerade noch gefehlt :
Unser bewährtes serielles Interface mit folgenden Features:

☐ Basicprogramme und Memobankdaten im ASCII-Format bidirektional übertragen.
☐ Abspeichern, editieren, ausdrucken über den ST
■ komplett anschlußfertig, incl. Software und Kabel für nur 119.- DM direkt bei:

E V E R Y W A R E
Jochen Flimm ☎: 0641-34232
Walltorstraße 41 D-6300 Gießen
PC-, Amiga-, C64-Version a. Anfr.

ST~TRESOR



ST-TRESOR kostet nur 69,-DM zzgl. 5,-DM Versandkosten.
Versand per Nachnahme durch:
F.Kepper
Lorettostraße 24
D-7800 Freiburg

neu

ST COMPUTER JAHRE

AKTUELLES

SOFTWARE

Allgemeines

BUSINESS (Datenbanken, Tabellenkalkulationen, Handel, Fibu)

DFÜ

GRAFIK / CAD

MUSIK

PROGRAMMIERSPRACHEN

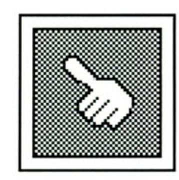
SINHALTSVERZEICHNIS

TEXTVERARBEITUNG / DTP

SOFTWARELISTINGS/ PROGRAMMIERPRAXIS

HARDWARE / PROJEKTE

HARDWARETEST



RELAX (SPIELE)

GRUNDLAGEN

SINHALTSVERZEICHNIS

ANWENDUNGEN

ST-ECKE

BÜCHER

Computer Dealers Exposition
COMDEX '88
in Las Vegas



Aus den umliegenden Bergen gelangt man in das Wüstental in Nevada, in dem Las Vegas liegt. Nachts erscheint es erst wie eine normale, mittelgroße Stadt, doch aus der Nähe betrachtet scheint ein Teil der Straßen ganz besonders gut erleuchtet, in wahre Lichtfluten getaucht. Diese Gegend ist das Casinoviertel am Las Vegas Boulevard, der auch den Beinamen *'The Strip'* trägt. In den riesigen Hotel- und Casinokomplexen und dem Las Vegas Convention Center, das ganz in der Nähe liegt, findet eine der großen amerikanischen Computermessen, die Herbst-COMDEX, statt. Vom 14. bis 18. November war Las Vegas mit etwa 150000 angereisten Fachbesuchern total überfüllt.

Wer die wahre Bedeutung des Wortes 'Reizüberflutung' nicht kennt, dem sei es dringend geraten, sich nachts, für beste Wirkung unvorbereitet und übermüdet, ins Casinoviertel (fast die ganze Stadt) von Las Vegas zu begeben. Überall blitzen Leuchtreklamen von unvorstellbaren Ausmaßen und entsprechender Geschmacklosigkeit. Das Wesen der Stadt spiegelt sich in der Messeorganisation wider. Die Messehallen der Stadt sind mit dem Ausstellerangebot nämlich bei weitem überfordert. Daher wurde ungefähr ein Drittel der Stände in die großen Hotels der Stadt ausgelagert. Besucher, die darauf angewiesen waren, mehrere Austellungsplätze zu besuchen, mußten viel Zeit und gute Nerven mitbringen. Die amerikanischen Händler, die im wesentlichen das Publikum dieser Fachmesse ausmachen, ertrugen es gelassen.

Die COMDEX ist, den Verhältnissen auf dem amerikanischen Markt angepaßt, im

wesentlichen eine PC-Messe. Natürlich waren auch die Peripherie-, Bauteil- und Zubehörhersteller entsprechend vertreten. Selbst Apples Einfluß war nur am Rande zu spüren. Sensationen? Eigentlich nicht. Der mit Spannung erwartete Rechner der neuen Firma von Apple-Gründer Steven Jobs, der NeXT-Würfel, war auf der Messe gar nicht zu sehen, nur eine Privatvorführung in einem Hotel wurde für ausgewähltes Publikum veranstaltet. Am wichtigsten für den Endanwender ist jedoch vielleicht die Tatsache, daß optische Speicherplatten mit Kapazitäten um die 600 Megabyte, die beliebig oft be- und überschreibbar sind, seit Herbst für die Hardwarehersteller zur Verfügung stehen. Zwei Beispiele: Sonys SMO-501-Laufwerk kann bis zu 650 Megabyte auf einer 5 1/4"-Kassette speichern, bei Zugriffszeiten, die nicht allzu weit über denen einer Festplatte liegen. Der Preis: 4.500,- Dollar bei Einzelstückzahlen. Wesentlich preisgünstiger dürften die Laufwerke von Canon sein, die jedoch 'nur' über 256-512 Megabyte Speicherkapazität verfügen. Alle diese Laufwerke lassen sich über ein Standard-SCSI-Interface mit dem Computer verbinden. Leider kann ein ATARI-Anwender mit dieser Technologie überhaupt nichts anfangen: Solange das TOS mit diesen Massenspeichergrößen nicht umgehen kann, müssen wir alle warten. Macht nichts, die Preise müssen sowieso noch etwas fallen.



Am letzten Messetag sorgte ein Auftritt der legendären Gruppe Fleetwood Mac für Aufsehen.

Damit wären wir bei ATARI angelangt; Sam Tramiel hat in Düsseldorf zwar ziemlich übertrieben, als er eine Halle von der Größe eines Fußballfeldes versprach, aber immerhin, ATARI verlor sich nicht im Gewühl der großen Messehallen, sondern war, privat und angenehm, in einem eigenen Saal untergebracht. Passenderweise trug dieser den Namen *'Gold Room'* - mit rotem Teppich ausgelegt, so dick, daß die Suche nach einem verlorenen Schlüssel eine Suchmanschaft erfordert hätte. Aber das nur am Rande.

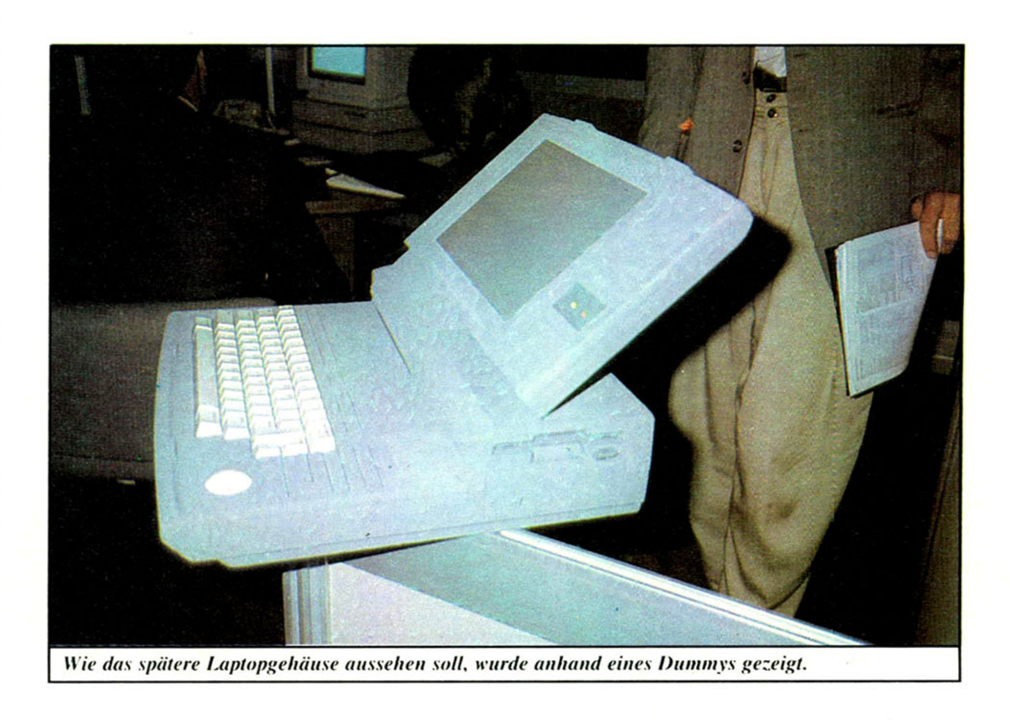
ATARI hat auf aufwendige Dekoration verzichtet und den Saal mit Ständen gefüllt, an denen verschiedene Softwarehersteller ihre Produkte vorführen konnten. Der amerikanische Markt ist für ATARI nach wie vor sehr schwach, weil nicht genug Geräte zur Verfügung stehen: umso erfreulicher war es, daß doch recht viele amerikanische Entwickler zu finden waren. Wer in Düsseldorf gewesen ist, hat aber wahrlich nichts versäumt: Die Größe der Ausstellung war im Vergleich zur ATARI-Messe doch recht bescheiden. Eine Bühne im Gold Room wurde praktisch den ganzen Tag von verschiedenen Softwarehäusern für die Vorführung von Produkten genutzt.

Ein besonderes Bonbon für die Liebhaber von 'Good old Rock 'n' Roll': *Fleetwood Mac* gab am Ende eines Messetages ein kurzes Konzert auf der Bühne des ATARI Saales. Das war die einzige Zeit, zu der der Gold Room wirklich voll war...

Sie fragen sich vermutlich, wieso immer nur von Software die Rede ist. Nun, ganz einfach: ATARI hatte keine Neuigkeiten für die Öffentlichkeit dabei. Der TT, die 68030-Maschine, die auch UNIX-fähig sein wird, ist noch nicht fertig. Es sind auch keine näheren Informationen zu dem Rechner verfügbar, weil ATARI einen totalen Informationsstop über alle neuen Produkte verhängt hat. Dies ist eine Folge der Indiskretionen bzgl. vertraulichen Materials über den TT: Ein deutsches ATARI-Magazin hatte ein ungefähres Blockschaltbild des TT, das nur für interne Benutzung bei ATARI gedacht war, veröffentlicht. Nun, die Konsequenzen bei ATARI sind deutlich: Man gibt sich höchst zugeknöpft.

ATARIs portabel ST

Hinter den Kulissen, bei einem ATARI-Aussteller-Fest, wurde jedoch auch ein Prototyp des Laptop gezeigt. Die Elektronik ist aber noch so groß, daß man getrost von einem Zwei-Mann-Portable



Wie das spätere Laptopgehäuse aussehen soll, wurde anhand eines Dummys gezeigt.

NEUE BÜCHER ZUM ATARI ST



über 530 Seiten
Bestell-Nr. B-419 **54,–**
ISBN 3-923250-69-X
Inclusive Programmdiskette

MERKMALE:
Forth ist, Kenner wissen das, eine der leistungsfähigsten Programmiersprachen für Micros. Sein Konzept der totalen Erweiterbarkeit eröffnet die Möglichkeit, diese Sprache entsprechend den eigenen Anforderungen beliebig zu erweitern, um zu dem Punkt zu kommen, wo selbst die komplexeste Anwendung nach außen nichts weiter als ein einziger neuer Forthbefehl ist. Dabei vereint Forth den Komfort einer Hochsprache mit den typischen Geschwindigkeitseigenschaften und der Universalität von Assembler.

Seit langem gibt es für den Atari ST das volksForth-83, welches obwohl Public Domain – die Leistungsfähigkeit vieler professioneller Entwicklungspakete bei weitem in den Schatten stellt. Es wurde entwickelt und wird unterstützt von der hochqualifizierten und engagierten Forthgesellschaft e.V.
Das vorliegende Buch ist Lehr- und Handbuch zugleich – die erste und bislang einzige ausführliche Referenz zu diesem hervorragenden Entwicklungssystem. Es gibt darüber hinaus tiefe Einblicke in die Interne des Systems.

AUS DEM INHALT:
- Einführungskurs in Forth
- Behandlung von Files
- Definitionen und Programmstrukturen
- Massenspeicher und File-Interface
- Forth – Assembler und Grafik



ca. 290 Seiten
Bestell-Nr. B-418 **59,–**
ISBN 3-923250-67-3
Inclusive Programmdiskette

MERKMALE:
Das Übungsbuch – CAD-Praxis – ermöglicht Ihnen einen problemlosen Einstieg in die zukunftsweisende Technik des computerunterstützten Zeichnens und Konstruierens mit dem Programm – CAMPUS.

Aufgrund der Praxiserfahrung hat es sich ergeben, grundsätzlich die Funktionen mit Beispielen darzustellen. Das Buchkonzept ist voll für ein Selbststudium ausgelegt.

Die ersten Übungsbeispiele testen das Zusammenspiel der Hard- und Software mit den Peripheriegeräten.

Mit dem Aufbau der wichtigsten Normkomponenten wird ein Grundstock geschaffen, der immer wieder Verwendung findet. Eingebaute Grafikbeispiele erhöhen den Lernerfolg und beschleunigen die Einarbeitung.

Das Buch schneidet viele Sachthemen aus den Gebieten Norm, Büroverwaltung, Elektronik, Maschinenbau, Ergonomie und Informatik an.

Einen weiteren Schwerpunkt stellt der Aufbau von Symbolbibliotheken für die Elektronik, Ergonomie und Informatik dar. Mit diesen Symbolen wird an Hand eines Beispiels ein funktionsfähiges Netzteil und ein Ablaufplan erarbeitet.
Mit dem Anhang, der den Buchteil abschließt, wird eine Fundgrube von praktischen Erfahrungen angeboten.

Programm-Diskette mit Übungs- u. Anwendungsbeispielen im Buch enthalten.



ca. 265 Seiten
Bestell-Nr. B-408 **59,–**
ISBN 3-923250-47-9

MERKMALE:
Wenn Sie das Software-Paket VIP-Professional kaufen wollen oder schon besitzen, dann weiht Sie dieses Buch schnell und umfassend in die Geheimnisse dieses Profiprogrammes ein.

VIP Professional besteht aus den drei Funktionsbereichen

- **DATENBANK**
- **KALKULATION**
- **GRAFIK**

mit denen wichtige und vielfältige Aufgaben hervorragend gelöst werden.

Das Buch ist neu überarbeitet und erweitert. Es enthält komplette Musterlösungen für die Gewinn- und Verlustrechnung und Fakturierung und weitere Beispiele.

Anhand dieser Beispiele wird gezeigt, wie Sie alles herausholen, was in VIP-Professional steckt.

Mit diesem Buch können Sie VIP-Professional richtig einsetzen und seine Möglichkeiten voll ausschöpfen.

Programm-Diskette im Buch enthalten.



über 330 Seiten
Bestell-Nr. B-420 **54,–**
ISBN 3-923250-70-3

MERKMALE:
Das Buch vermittelt neben dem nötigen Hintergrundwissen praktische Anwendungen in GFA-BASIC (ab Version 2.0).

Es werden nur geringe oder gar keine Vorkenntnisse in BASIC benötigt. Wichtige Befehle werden ausführlich erläutert.

Die einzelnen Prozeduren fügen sich zusammen zu einem Programm, das nach eigenen Bedürfnissen gestaltet werden kann.

Komplette Listings werden zusätzlich mit Struktogrammen erläutert.

AUS DEM INHALT:
- Der Weg zum Bohrschen Atommodel zum modernen Orbitalmodel
- Wie Probleme in Programme umgesetzt werden
  z. B. ein Programm zur Bestimmung der Elektronenkonfiguration eines Elementes
- Der Aufbau des Periodensystems der Elemente
- Modelle für die chemische Bindung (Atom, Ion, Metallbindung)
- Periodische Eigenschaften und die Gruppen des Periodensystems
- Einführung in die Stöchiometrie. Grundbegriffe der Stöchiometrie: Mol, Molmasse...

Programm-Diskette mit Übungs- und Anwendungsbeispielen im Buch enthalten.

* alle Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir:
_______St. **Das große VIP-Buch** á DM 59,–
_______St. **CAD – Praxis mit Calamus** á DM 59,–
_______St. **Einführung in volksForth-83** á DM 54,–
_______St. **Chemie – Lernen mit dem Computer** á DM 54,–

zuzügl. Versandkosten 5,– DM (unabhängig von der bestellten Stückzahl)

Name, Vorname ______________________

Straße, Hausnr. ______________________

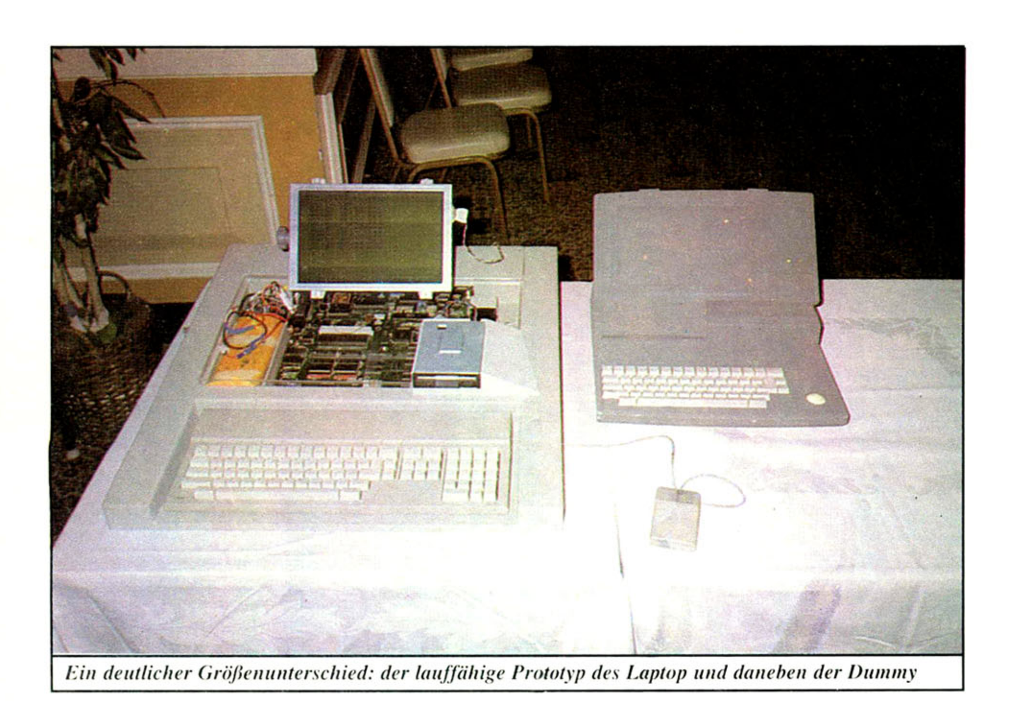
PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

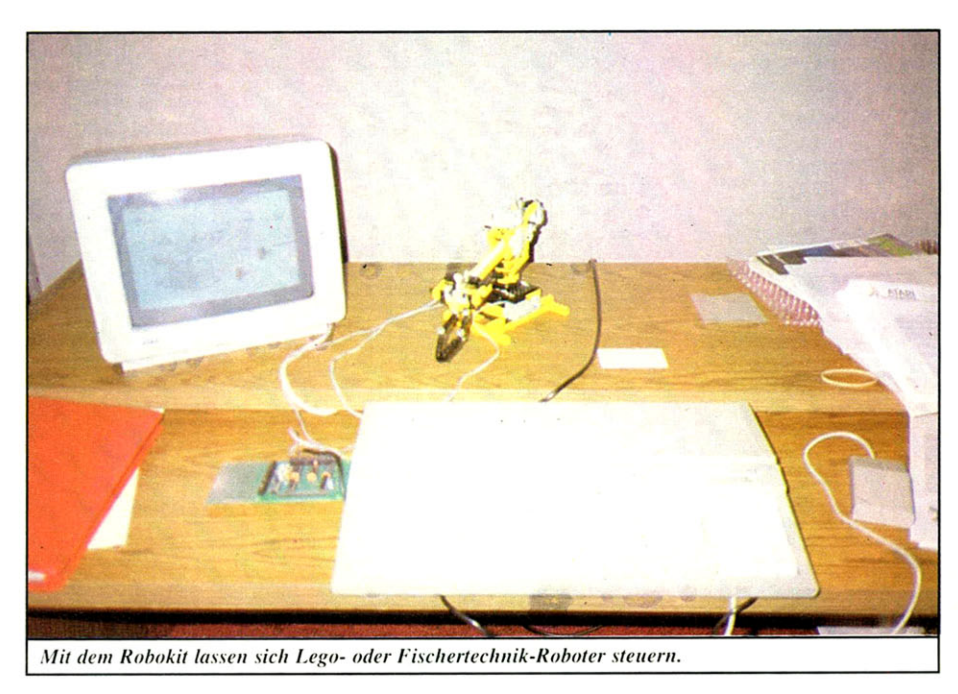
SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich



Ein deutlicher Größenunterschied: der lauffähige Prototyp des Laptop und daneben der Dummy

(für **A**TARI **T**ransputer **W**orkstation; so heißt der Rechner jetzt - Abaq ade) so richtig austoben. Ein Beispiel: Fenster werden nicht wie auf dem Mac oder unter GEM als Geisterrahmen verschoben, sondern vollständig, komplett und mit Inhalt. Das Ganze geht sogar ziemlich ruckfrei vor sich. Mit der neuen Version kann man jetzt richtig arbeiten. Es gibt noch mehr Software-News: Eine neue Version des Helios-C-Compilers und ein C-Source-Level-Debugger, der noch vor Weihnachten in den Beta-Test gehen soll. Schließlich hat Prospero Software eine Helios-Version des bekannten Pro-Pascal-Compilers vorgestellt. Auch Modula II, Fortran und BASIC sind ab Januar verfügbar, ebenso das INMOS Transputer Development System in einer Helios-kompatiblen Version.

sprechen kann. Das Display (640*400 Punkte) sieht dafür aber hervorragend aus, mit blauem Hintergrundlicht. Bei dem Prototyp handelte es sich um einen umgebauten Mega ST , der mit CMOS-Chips und einem LCD-Display ausgestattet immerhin ca. zwei Stunden im Batteriebetrieb arbeitete. Auch ein Gehäuseprototyp wurde gezeigt: sehr schick. Das wird ein richtiger Edel-Laptop, mit einem Trackball statt Maus und einem eingebauten 720 Kbyte-3,5"-Laufwerk an der rechten Seite. Es ist aber kaum damit zu rechnen, daß das Gerät vor Mitte nächsten Jahres zu haben sein wird. Auch über die Ausstattung wurde Stillschweigen bewahrt - bisher ist nicht entschieden, wieviel Speicher der Laptop haben wird. Mit Sicherheit gibt es aber eine Option für eine 20MB-Harddisk.



Mit dem Robokit lassen sich Lego- oder Fischertechnik-Roboter steuern.

Das CD-Rom war überhaupt nicht zu sehen, vielleicht eine Konsequenz der ATARI-Messe, wo man sich mit einem Laufwerk ohne Steuersoftware eher blamierte. ATARI versichert allerdings, daß die Hardware fertig und auslieferungsbereit ist. Aber was soll ein Käufer mit einem Laufwerk ohne Daten und Software...

Apropos Laufwerk: Statt der bisherigen 20MB-Festplatte soll in Zukunft ein 30-MB-Laufwerk in den ATARI-Harddisks eingebaut sein. Und das zum gleichen Preis.

Eher eine kleine Neuheit ist der Robokit. Dabei handelt es sich um ein Interface für die Steuerung von Servomotoren, Schaltern und Sensoren sowie eine entsprechende Steuerungssoftware. Aus Fischertechnik oder Lego (oder ähnlichen Baukästen) können nun funktionsfähige Roboter gebaut werden.

Echte Hardwareneuigkeiten gab es eigentlich nur in Sachen Transputer. Die Vorseriengeräte (Release 4), von denen zur Zeit 500 Stück bei ATARI Braunschweig gebaut werden, waren, in ein großes quadratisches Towergehäuse verpackt, zu sehen. Intern hat sich nicht viel geändert, außer daß einige ICs zu Custom-Chips zusammengefaßt wurden. Auch ist kein externer Mega-ST mehr notwendig; eine dem Mega-ST entsprechende I/O-Platine ist eingebaut.

Auch auf der Softwareseite hat sich einiges getan: Eine neue *X Windows*-Version darf sich mit dem neuen Blitter des *ATW*

Software-News

Drei Softwareschwerpunkte waren auf der Messe zu verzeichnen: Desktop Publishing & Textverarbeitung, Business-Anwendungen und Midi.

Inzwischen ist ein wirklich umfangreiches DTP-Angebot für den ST erhältlich. Drei wirklich große und umfangreiche DTP-Pakete der Qualitätsklasse von Calamus waren auf der Messe zu sehen: *Calamus* selbst, dann der Nachfolger des Publishing Partners, der ursprünglich Publishing Partner professional heißen sollte, jetzt aber den Namen *PageStream* trägt, sowie ein von ATARI selbst vertriebenes Paket mit Namen *DeskSet II*. Alle diese Programme sind übrigens in der Lage, mit einem 19"-

Bildschirm zusammenzuarbeiten.

PageStream scheint, soweit man es nach einer kurzen Vorführung sagen kann, tatsächlich zu halten, was der alte Publishing Partner versprach. Das Programm ist voll PostScript-kompatibel, der Weg zu professionellen Ausgabemedien steht also offen. Selbst Farbdruck oder separate Farbauszüge sind möglich. Zehn Fonts werden mitgeliefert, alle Möglichkeiten der Textverarbeitung sind vorhanden, natürlich auch Trennhilfe, Rechtschreibprüfung usw. Einfache objektorientierte Grafik kann im Programm erzeugt werden, ebenso lassen sich Bilder mit 300dpi-Scannern einlesen (und natürlich auch von allen gängigen Grafikprogrammen). Texteffekte gibt es in Hülle und Fülle, weit über einfachen rotierten Text hinausgehend. Text kann problemlos auch um unregelmäßig geformte Grafiken herumfließen. Man darf auf eine deutsche Version gespannt sein.

Über DeskSet waren leider noch nicht so viele Informationen erhältlich, auch dieses Programm macht aber einen sehr leistungsfähigen Eindruck. Besonderes Kennzeichen: Direkter Anschluß an CompuGraphic-Satzbelichter ist möglich. das Programm verwendet sogar Original-CompuGraphic-Fonts! Allerdings ist dies für den amerikanischen Markt wichtiger, da die CompuGraphics-Belichter in Europa weit weniger verbreitet sind als in den USA.

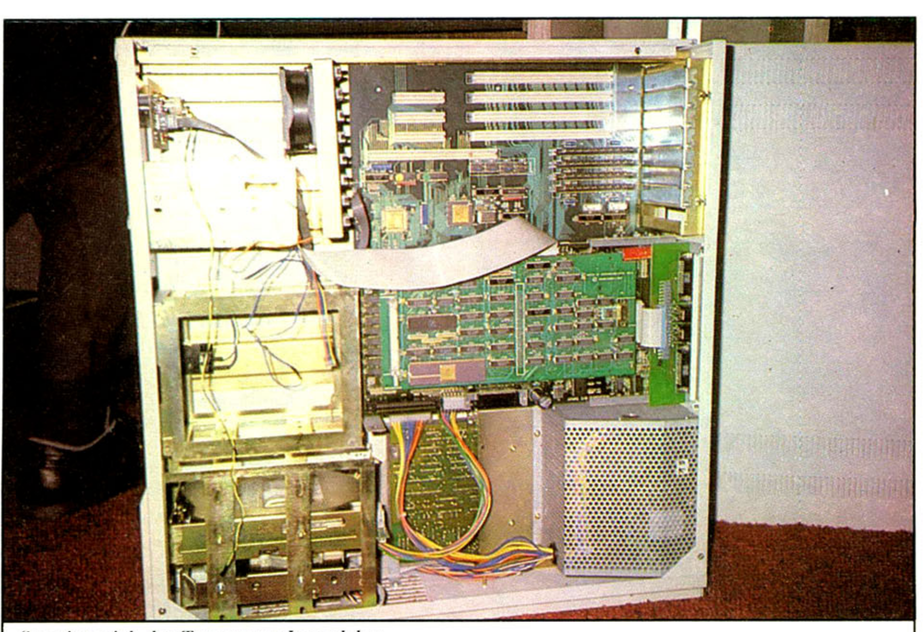
Die Layoutfähigkeiten dürften in etwa denen von PageStream entsprechen, einige zusätzliche Besonderheiten wie Trennregeln für acht verschiedene Sprachen sind auch eingebaut. Die Bedienung von DeskSet II wirkte sehr komfortabel.

Textverarbeitung

Nicht nur bei den Layout-Programmen, auch im Bereich der 'einfachen' Textverarbeitung tut sich etwas. Die Beta-Testversion von WordPlus 3.0 ist ja schon seit einiger Zeit im Umlauf, aber es scheint endlich ernstere Konkurrenz zu geben.



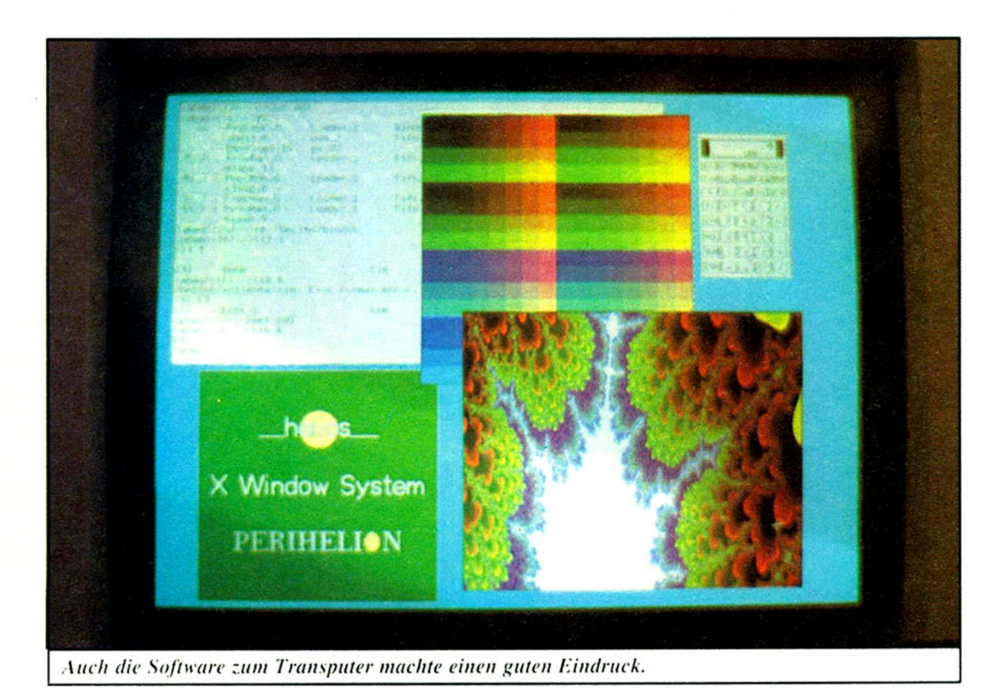
Der ATARI-Transputer ATW (ehemals Abaq) zeigte sich im neuen Towergehäuse.



So zeigte sich das Transputer-Innenleben.

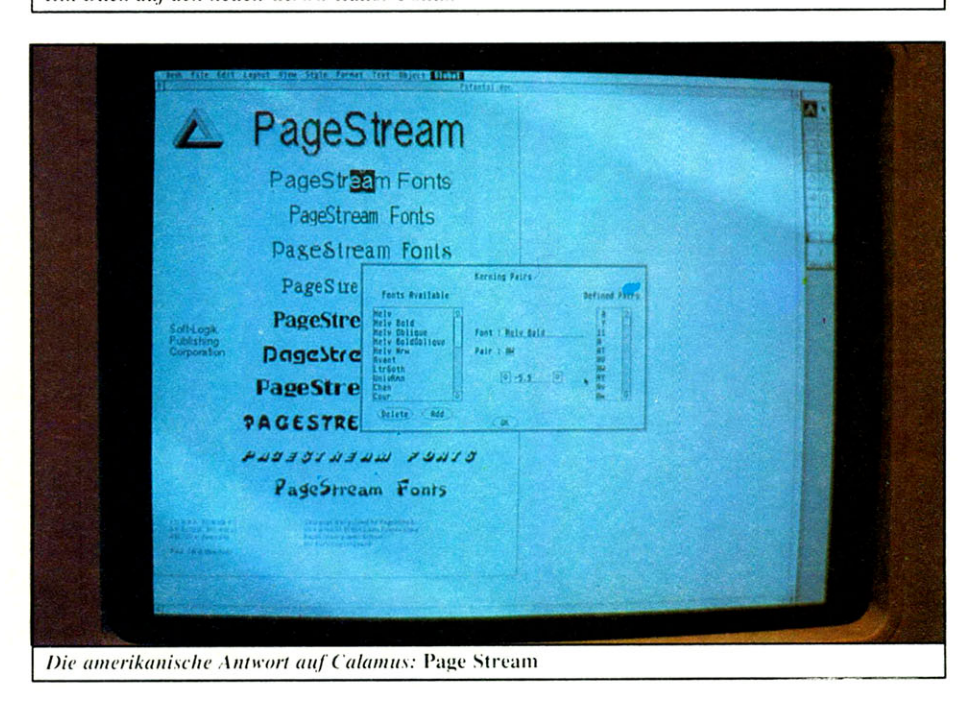
Microsoft und Word Perfect, der Welt größte Hersteller von Textverarbeitungen, scheinen ihre ST-Aktivitäten doch etwas forciert zu haben. Zumindest war auf der Messe eine Anpassung von *Microsoft Write* zu sehen, die, im Gegensatz zu den Versionen, die voriges Jahr Deutschlands Raubkopierer schockierten, auch richtig zu funktionieren schien. An eine Anpassung des PC- und Mac-Dauerbrenners *MS Word* scheint man aber nicht zu denken. Dafür ist aber die neueste ST-Version von *Word Perfect*, die einer PC-Version irgendwo zwischen 4.1 und 4.2 entspricht, recht vielversprechend. Bei der Vorführung stürzte das Programm nicht ab, es wirkte sehr stabil und sicher, die Fähigkeiten von Word Perfect sind ja bekanntermaßen über jeden Zweifel erhaben. Leider haben die deutschen Versionen für Mac und PC bei Word Perfect Vorrang, so daß man wohl noch ein wenig auf Word Perfect ST in deutscher Fassung warten muß. Bleibt nur zu hoffen, daß nicht allzu viele Benutzer von den bisherigen ST-Versionen (katastrophal ist eine harmlose Umschreibung) von Word Perfect abgeschreckt wurden. Wie der Word Perfect-Mitarbeiter auf der Messe erklärte, habe man inzwischen auch eingesehen, daß die ST-Version nicht so teuer sein kann, wie die Versionen für andere Rechner. Hoffentlich schließt sich die deutsche Abteilung dieser Auffassung an, dann gibt es bald eine wirklich brauchbare Textverarbeitung auf dem ST.

Eine reine Textverarbeitung, die unverdienterweise ein Mauerblümchendasein fristet, ist *REGENT WORD II*. Das Programm ist extrem billig (25 $) und wirklich brauchbar. Allerdings ist es meines Wissens in Deutschland nicht erhältlich.



Auch die Software zum Transputer machte einen guten Eindruck.

Ein Blick auf den neuen GDOS-Editor Fontz!



Die amerikanische Antwort auf Calamus: Page Stream

Wordflair & WordUp

Ein sehr interessantes Programm für Textschreiber ist *Wordflair* von Blue Chip international. Wordflair ist kein DTP-Programm und keine Textverarbeitung. Wordflair liegt, im positiven Sinne, irgendwo dazwischen. Der Hersteller bezeichnet es als '*Document Processor*'. Man kann Texte schreiben und formatieren, grafische Gestaltungsfunktionen sind auch vorhanden. Zusätzlich enthält Wordflair noch Elemente einer Tabellenkalkulation und Business-Grafik. Schließlich ist das Ganze noch sehr leicht zu bedienen. Einziger Nachteil: Es ist noch nicht fertig.

WordUp (von Neocept) ist eine Textverarbeitung, die ebenfalls einige Elemente von DTP-Programmen enthält. So werden verschiedene Fonts auf dem Bildschirm unterstützt, auch das Seitenlayout kann sehr flexibel gehandhabt werden. Ähnlich wie der Timeworks Desktop-Publisher arbeitet es mit GDOS-Fonts. Grafikeinbindung ist selbstverständlich. Außerdem ist das Programm sehr schnell.

Utilities für Textverarbeitung und DTP

Der SLM 804 Laserdrucker wird Post-Script-fähig! '*UltraScript*' heißt der PostScript-Emulator. Das Programm soll von ATARI vertrieben werden und sofort erhältlich sein. Mitgeliefert werden 16 Fonts, separat werden zwei weitere Fontsätze, die alle Originalfonts des Apple-Laserwriters II enthalten, erhältlich sein.

'*Fontz!*' heißt ein sehr leistungsfähiger GDOS-Fonteditor, der von der gleichen Herstellerfirma wie WordUp kommt. Hochspezialisierte Zeichenfunktionen machen das Layout von Buchstaben erheblich einfacher, als man es gewohnt ist.

Vom gleichen Hersteller kommt Turbo-Jet, ein Druckertreiber für Hewlett Pakkard Tintenstrahl- und Laserdrucker. Mit dem Treiber können die HP-Drucker mit jedem GDOS-Programm zusammenarbeiten, und das sogar mit beachtlicher Geschwindigkeit.

Midi

Zwar waren zahlreiche Midi-Softwarehäuser auf der Messe vertreten, und auch bei den Softwarepräsentationen auf der Bühne nahmen Midi-Programme breiten Raum ein, aber wirkliche oder sensationelle Neuigkeiten gab es dennoch nicht zu



Page Stream bietet Möglichkeiten, die für ein DTP-Programm ungewöhnlich sind.

sehen. Im wesentlichen wurde das bekannte Programm der Hersteller vorgeführt.

Erwähnenswert war jedoch ein französisches Programm namens *Amadeus ST*. Hierbei handelt es sich um ein recht umfangreiches Musiklernprogramm, das einen großen Theoriebereich gründlich abdeckt und auch für Kinder gut geeignet ist. Eine deutsche Version ist in Vorbereitung, leider hat der Hersteller aber eine sehr hohe Preisvorstellung (ungefähr 500,- DM).

Geschäftliches

Wer schon immer dBase III+ vermißt hat, kann sich freuen: Die neueste dBman-Version, *dBman V*, ist vollständig dBase II und III+ kompatibel und außerdem schneller. Fertig programmierte Applikationen können außerdem mit einem '*greased Lightning*' (geölter Blitz) getauften Compiler übersetzt werden, um maximale Ausführungsgeschwindigkeit zu erreichen. Als Besonderheit läßt sich dBman mit der Maus über Pulldown-Menüs, die auch selbst programmiert werden können, bedienen.

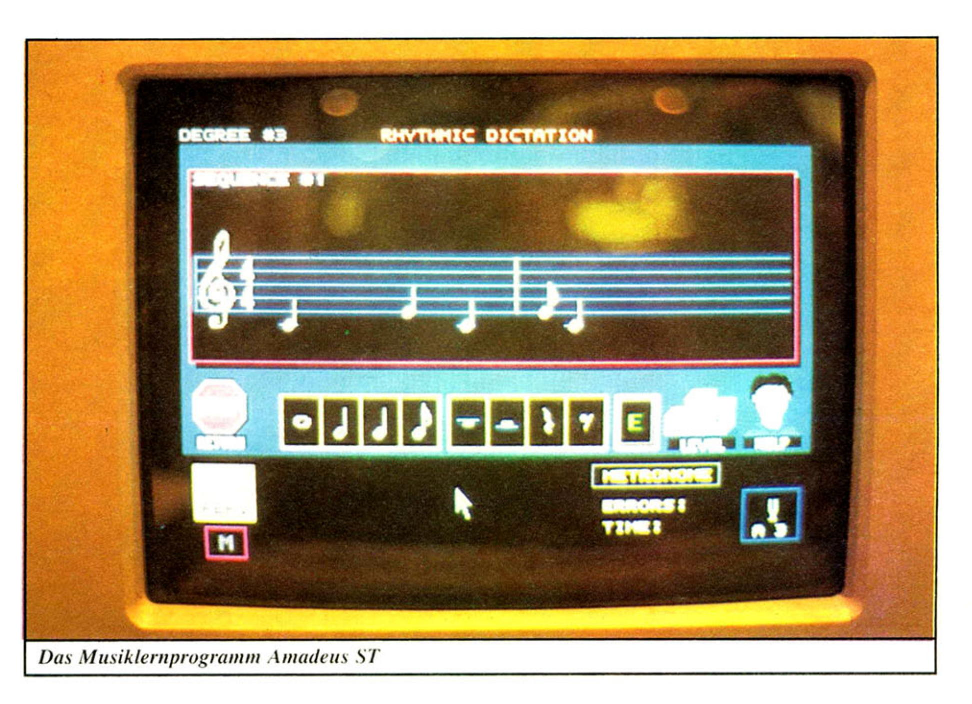
SBT aus Kalifornien liefert gleich eine ganze Palette von dBase-Anwendungen passend zu dBman V. Leider sind diese Geschäftsanwendungen vollständig auf den amerikanischen Markt zugeschnitten.

Die bekannte Datenbank *Superbase professional* ist jetzt in ihrer Version 3.0 lieferbar. Als Erweiterung wurde eine Telekommunikationsfunktion integriert, der Formulareditor wurde stark verbessert.

LDW hat ein, wie der Name '*Power*' schon suggeriert, extrem leistungsfähiges Spreadsheet im Programm. Der Leistungsumfang ist an Microsofts Excel angelehnt. Ein exakter Excel-Clone ist bei ATARI aber auch schon in Arbeit und wird in Kürze lieferbar sein.

Eine ganz besondere ST-Anwendung ist *ViewTouch*. ViewTouch ist eine Komplettlösung, die aus Hardware, Software und Schulung besteht. Kern des Systems ist ein Monitor, der auf Fingerdruck reagiert. Damit können Kassen zum Beispiel für Restaurants oder Ladenketten konzipiert werden, die extrem unkompliziert in der Bedienung sind und sich an jede Anwendung anpassen lassen. Die Angestellten müssen nur noch die entsprechenden Felder auf dem Bildschirm antippen, verschleißanfällige und unflexible Tastenfelder werden nicht mehr gebraucht. Die Herstellerfirma paßt jedes ViewTouch-System komplett an die Wünsche und Bedürfnisse des Kunden an.

Gleich mehrere Hersteller hatten Komplettlösungen für Lagerhaltung und Verkaufsabrechnung im Programm. Einige dieser Lösungen sind mit Bar-Code-Leser und ST-gesteuerter Kassenschublade ausgestattet und ermöglichen so die komplette und höchst komfortable Kontrolle des Warenflusses von Einkauf bis Verkauf. Leider sind derartige Anwendungen bisher nur für den amerikanischen



Das Musiklernprogramm Amadeus ST

Markt geeignet, an Anpassungen für Deutschland wird aber gearbeitet.

Grafik

Auch im Bereich Computer & Grafik gab es einiges zu sehen. Die Produkte der Firma ANTIC waren weit verbreitet, auf zahlreichen Monitoren liefen Animationen, die mit der 2D/3D-Animationsserie *Cyber-Studio* erzeugt waren. Für Cyber-Studio gibt es einige Neuerscheinungen: Ein neuer Objekteditor namens *Cyber-Sculpt* bietet extrem vielseitige Funktionen für die Erzeugung von 3D-Objekten. Manche dieser Funktionen gibt es sonst nur auf großen Grafik-Workstations. Da Cyber-Studio von einigen US-Computergrafik-Studios für die Vorbereitung von Animationen verwendet wird, wurde ein Filekonverter angekündigt, der CAD 3D- (oder Cyber Sculpt-) Objekte in Objekte für professionelle Animationssysteme umwandelt.

Ein weiteres Programm der Cyber-Serie erlaubt es, 3D-Objekte in beliebige ST-Bilder 'einzupacken'. Damit kann man zum Beispiel eine Coladose oder ähnliche Objekte mit 'bemalter' Oberfläche erzeugen.

Im Bereich CAD war der auffälligste Teilnehmer die ST-Version von *Drafix*, das auf dem IBM bereits bekannt ist. Die ST-Version gleicht der PC-Fassung bis ins Detail. Ansonsten waren das bereits auf der ATARI-Messe in Düsseldorf vorgestellte 2D/3D-CAD-Programm *Dyna-Cadd* und *BeckerCad* zu sehen.



Zusammen mit einer Polaroid Palette kann Neriki *Monitorbilder auf Dias und Fotos zaubern.*

Die Herstellerfirma von Easy-Draw, Migraph, hat ein Zeichenprogramm für hochauflösende Bit-Map-Grafiken, wie sie für DTP-Zwecke benötigt werden, angekündigt. In diesem Programm werden sehr leistungsfähige Zeichenwerkzeuge enthalten sein, so z.B. für *Bezier-Kurven* oder *B-Splines*. Es wird den Namen 'Touch-Up' tragen.

Die französische Firma Human Technologies zeigte das Zeichenprogramm *ZZ-Lazy Paint*, das durch besonders gute Ausdruckqualität bis zu 300dpi (neben höchst leistungsfähigen Zeichenfunktionen) überzeugt. Es ermöglicht auch PostScript-Ausgabe und ist somit prädestiniert für DTP-Anwendungen, z.B. mit PageStream.



Das JRI-Genlock-Interface für "nur" 500 $

Sprachen

In diesem Bereich gibt es nur eine echte Neuheit: Einen richtigen *Cobol*-Compiler für den ST. Für 199 Dollar erhält man einen ANSI 74-kompatiblen Compiler der Firma CASTECH Software Systems , der echten 68000-Code erzeugt, einen GEM-Editor und eine Shell. GEM-Routinen können von Cobol aus aufgerufen werden.

Kommunikation

Bei Intersect Software war *Masterlink* als Steigerung des Kommunikationsprogramms Interlink zu sehen. Unter anderem wurden die Editorfunktionen sowie der Mailboxteil schneller und erheblich erweitert. Laut Aussage von Intersect soll Masterlink in Deutschland als *Interlink 2.0* zur CeBIT ´89 über Bela Computer zu haben sein.

Hardware

Eine sehr interessante Hardware-Erweiterung für Grafik-Freaks: Das *JRI-Genlock-Interface* für Mega-STs. Für nur 500 $ (!) erhält man eine Einsteckkarte für den Mega und einen Überblendregler, mit dem man zwischen Video- und Computerbild überblenden kann. Das gemischte Ausgangssignal ist ein echtes RS-170-Standard-Signal, das jeder NTSC-Videorekorder verarbeiten kann. Eine PAL-Version ist in Vorbereitung, ebenso eine externe Ausführung für die 'kleinen' ATARI STs.

ST Scan von Navarone ist ein Interface samt Software für Scanner der Canon-Serie IX-12. Die Auflösung beträgt bis

300dpi, die Scanzeit ist mit unter 15 Sekunden sehr kurz.

ICD bietet ein sehr komfortables *Tape-Backup-System* für ATARI-Harddisks an. Dabei werden nicht nur TOS-, sondern auch Magic Sac (Mac-Emulator) oder MS-DOS-Partitions verwaltet. Auf ein Band passen 155 Megabyte.

Die Polaroid Palette-Filmausgabeeinheit kann mit einer Software der Firma *Neriki* jetzt auch am ST für perfekte Farbhardcopies sorgen. Auch Bilder mit mehr als 16 Farben sind mit dem Palette-Rekorder möglich. Das System wird in Deutschland über ATARI vertrieben werden.

Einen sehr preisgünstigen Multisync-Farbmonitor stellte die Firma Wuztek vor. Der *Omnimon GS* kostet nur knapp über 500$ und bietet angeblich eine dem Multisync II vergleichbare Bildqualität.

Ein neuer Mac-Emulator mit Harddisk und Mac plus-ROMs

Spectre 128 heißt der Mac-Emulator der Firma 'Gadgets by Small'. Als bisher einziger Mac-Emulator ist dieses Gerät in der Lage, die 128KByte ROMs des Mac-plus zu benutzen und somit auch Hypercard und das HFS zu verwenden. Selbst der Multifinder läuft, allerdings, wie der Hersteller zugibt, nicht sehr stabil. Alle Finder-Versionen, die Apple jemals veröffentlicht hat, laufen unter Spectre. Harddisk-Support ist selbstverständlich, mit einer Geschwindigkeit, die laut Benchmarktests sogar die des Mac II übertrifft! Mit der Translator-Hardware des Magic Sac (sehr langsam), die zwischen Floppylaufwerk und Computer geschaltet wird, können Mac-Disketten direkt gelesen werden. Allerdings ist eine eigene Hardware für Mac-Diskformat in Vorbereitung.

Neue Version des MS-DOS-Emulators PC ditto

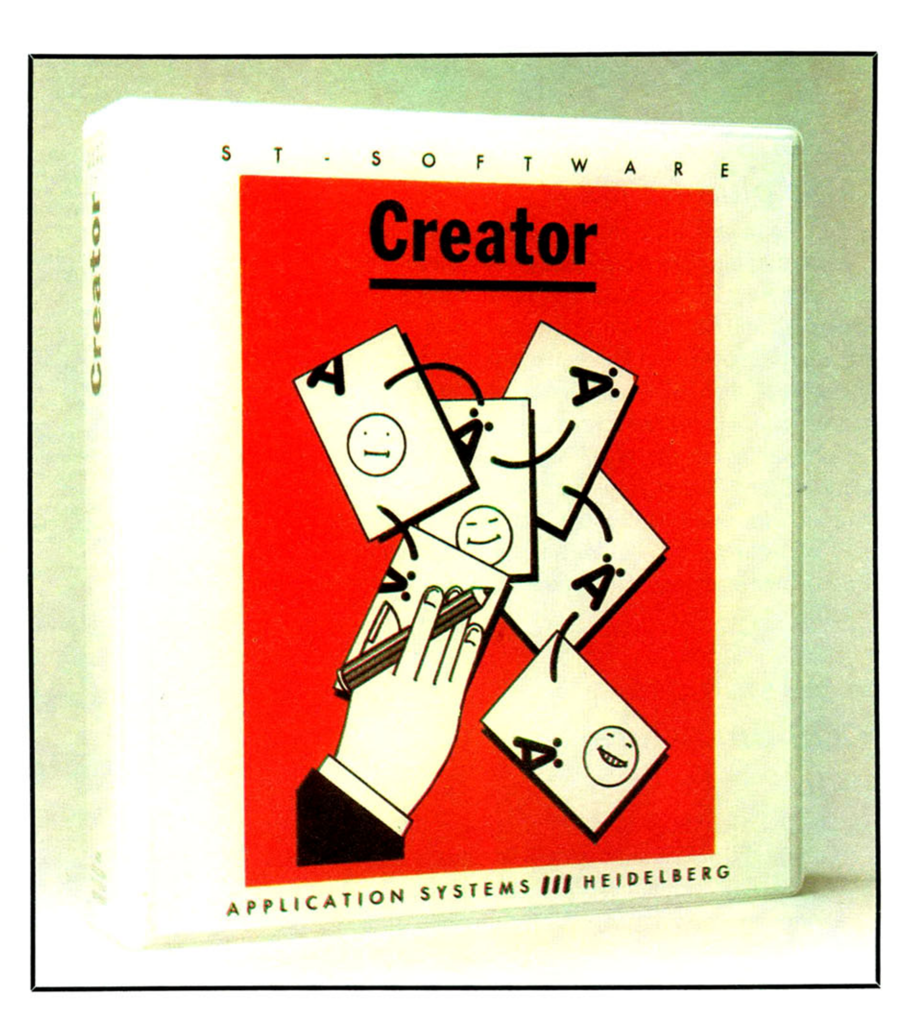
Nicht in der ATARI-Halle sondern in einem der unzähligen Hotels zeigte die Firma Avant-Garde Systems die neue *Version 3.96* ihres MS-DOS-Emulators *PC ditto*. Hauptmerkmale der neuen Version sind eine bessere Unterstützung verschiedener Festplatten und der Tastatur. Ferner konnte die Kompatibilität zu einem Original-MS-DOS-PC noch weiter gesteigert werden. Ein Gespräch mit Bill Teal, dem Geschäftschäftsführer von Avant-Garde Systems ergab, daß auch bisher nicht bei der MAXON Computer (für Europa) registrierte Kunden oder Kunden einer älteren Version sich in einem befristeten Zeitraum bei der MAXON Computer GmbH registrieren lassen können und somit preisgünstig in den Besitz von zukünftigen Versionen kommen können.

CS/UB

ENDE

THE CREATOR

Zu bewundern war der Creator im Entwicklungsstadium schon auf diversen Messen. Nach fast zwei Jahren ist er nun endlich fertig, und man kann sagen, daß sich das Warten gelohnt hat. Aber für alle Nichteingeweihten sei schon mal erwähnt, daß der Creator ein Zeichenprogramm ganz besonderer Art für hohe Auflösung (monochrom) ist. Neben allen möglichen (und unmöglichen) Bearbeitungsmodi ist es mit ihm auch möglich, die erzeugten Grafiken in Bewegung zu setzen, denn der Creator verfügt über einen Zeichen- und einen Animationsteil. Doch alles der Reihe nach.



Nach dem Laden befindet man sich im Zeichenteil. Es fällt zunächst einmal auf, daß die GEM-Menüleiste fehlt, und damit Accessories leider nicht erreichbar sind. Dafür sind aber fast alle Zeichengeräte und Manipulationsfunktionen direkt über eine Bedienungsleiste mit Icons am unteren Bildschirmrand aufrufbar. Um die Bedienung einfach und übersichtlich zu halten, sind viele Piktogramme mit artverwandten Funktionen übereinandergelegt. Es mag sich vielleicht kompliziert anhören, die Bedienung erweist sich aber als äußerst einfach und gelungen. Die Umschaltung auf andere Funktionen erfolgt ganz einfach durch Anwählen mit der rechten Maustaste, der Aufruf mit der linken. Damit die Iconleiste nicht beim Arbeiten stört, kann man einen sogenannten Swapmodus einstellen, bei dem sie automatisch ausgeblendet wird, wenn der Mauszeiger sie verläßt. Für maßgerechtes Arbeiten lassen sich auch am linken und oberen Rand des Bildschirms Lineale einblenden. Zusätzlich wird die aktuelle Mausposition in einem wählbaren Maßstab eingeblendet.

Bildformate

Die ersten beiden Icons sind die wohl wichtigsten überhaupt. Es sind die zwei Diskettensymbole, die es ermöglichen, Bilder zu laden, und die in harter Arbeit erstellten Kunstwerke der Nachwelt zu erhalten. Wenn es ums Laden geht, stellt sich für den Benutzer die Frage, ob auch mit anderen Programmen erzeugte Bilder gelesen werden können, schließlich hat man in seine Bildersammlung schon viel Arbeit gesteckt, die man natürlich nicht noch einmal aufwenden möchte. Um den Anwender auf keinen Fall zu enttäuschen, sind alle möglichen (und unmöglichen) Bildschirm- und Ganzseitenformate ladbar (s.Tabelle 1). Dabei wird auch vor Rechnergrenzen nicht haltgemacht, so können z.B. IFF-Format-Bilder vom AMIGA gelesen werden.

Der Creator ist beim Laden nicht auf ein Bildschirmformat von 640*400 Pixeln beschränkt, er liest auch größere Bilder in den Speicher und unterstützt maximal bis zu 10000*10000 Pixel, vorausgesetzt, es steht genügend Speicher zur Verfügung. Die Größe eines Ganzseitenbildes stellt Creator gegebenenfalls beim Laden automatisch ein. Sie läßt sich aber im Nachhinein beliebig vergrößern bzw. verkleinern, und bildschirmgroße Bilder (also 640*400 Pixel) lassen sich an eine belie-

Bildschirmformate

Extension	Art
IMC	SIGNUM! (*)
PAC, SEQ etc.	STAD (*)
IM...	IMAGIC (alle Auflösungen) (*)
DOO	Doodle (*)
PIC	GST englisches Word Plus (*)
IMG	GEM-Imageformat, IBM, Word Plus (*)
IFF	AMIGA-Standardformat (*)
PI...	DEGAS (alle Auflösungen) (*)
PC...	DEGAS komprimiert (alle Auflösungen)
ART	Art Director
NEO	Neochrome

Ganzseitenformate

Extension	Art
IMG	GEM-Imageformat, IBM, Macintosh (*)
SCN	Hawk-Scanner (*)
GEM	Metafiles (objektorientiert)
IMF	SIGNUM! (Fullscreen) (*)
BI1, BI4	Basis PC FAX

Die mit einem Stern (*) versehenen Formate können auch geschrieben werden.

Tabelle 1: Diese Formate können vom CREATOR geladen werden.

bige Stelle des Ganzseitenbildes hinzuladen (auf diese Weise können auch Bilderfolgen geladen werden.). Insgesamt kann Creator maximal 100 Normalbilder verwalten.

Sind nun die Bilder im Speicher, geht es an die ersten Mal- und Bearbeitungsversuche, wozu der Bildschirm am besten eingefroren wird. Einfrieren bedeutet, daß der beim Aktivieren der Funktion aktuelle Bildinhalt in einem Zwischenspeicher gesichert wird und jede darauffolgende Änderung somit wieder rückgängig gemacht werden kann. Das Einfrieren stellt somit eine einfache Möglichkeit zur Manipulation der eigenen Schöpfungen dar, ohne bei mißglückter, künstlerischer Tätigkeit den alten Bildinhalt umständlich über Neuladen von Diskette oder mühsam restaurieren zu müssen. Ist ein Mißgeschick passiert, wird der Schwamm aus der Iconleiste gewählt und der ganze Bildschirm oder nur Teile davon gelöscht. Schon kommt das ursprüngliche Bild wieder zum Vorschein. Zum Löschen sei bemerkt, daß es zwei Arten gibt. Die herkömmliche, wobei alles unter dem "Schwamm" gelöscht wird, und die zweite, bei der sich mit der rechten Maustaste eine Box aufziehen läßt, unter der alles gelöscht wird, sobald man die Maustaste losläßt. Es ist somit möglich, sehr schnell große Teile des Bildschirms zu leeren.

Aber man muß nicht immer gleich den Schwamm zücken und versuchen, mißglückte Bilder und Zeichnungen mühsam zu restaurieren oder pausenlos zwischen Einfrieren und Loseisen umzuschalten. Creator bietet ebenfalls eine UNDO-Funktion, die sich wahlweise bis auf vier Ebenen einstellen läßt. Das heißt, daß bei versehentlich zerstörtem Bildschirminhalt maximal die vier letzten Versionen wieder hergestellt werden können.

Gut gemalt, ist...

Die Standardzeichengeräte wie Pinsel, Stift, Sprühdose sind selbstverständlich auch im Creator vertreten. Es lassen sich, wie eigentlich in jedem einigermaßen guten Grafikprogramm, Linienstärke und -art beeinflussen. Creator bietet zwei Lupengrößen (Vergrößerungsfaktor 4 und 8), mit denen punktgenaues Arbeiten möglich ist. Nach dem Aufruf der Lupe erhält man ein "Lupenglas", wie man es bereits von STAD kennt, mit dem man sich den eigentlichen Lupenarbeitsbereich wählen kann. Es stehen allerdings auch einige nicht alltägliche Funktionen zur Auswahl. Der Pinseleditor erlaubt die Gestaltung von individuellen Pinselformen. Natürlich kann auch zwischen verschiedenen, vordefinierten Arten von Pinseln und Sprühdosen gewählt werden. Normalerweise hinterläßt ein Pinsel, egal wie schnell er bewegt wird, an jeder Stelle des Bildschirms seinen Pinselstrich, und eine Sprühdose sprüht ständig, solange die Maustaste gedrückt bleibt. So kennt man es auch aus anderen Zeichenprogrammen. Schaltet man aber das Pinselsymbol mit der rechten Maustaste um, arbeitet der Pinsel wie ein Kohlestift - man kann also mehrmals über dieselbe Stelle malen, bis eine vollständige Schwärze erreicht wird. Die umgeschaltete Sprühdose setzt nur dann Punkte auf den Bildschirm, wenn die Maus mit gedrückter Taste bewegt wird. Nichts Besonderes, möchten Sie sagen? Vielleicht, aber die Summe solcher Kleinigkeiten ergibt eben das gewisse Etwas, das ein Zeichenprogramm braucht, um auf dem Markt Gefallen zu finden.

Als Linienfunktionen stehen zunächst einmal die üblichen zur Verfügung, die man aus anderen Zeichenprogrammen kennt. Darüber hinaus stellt Creator zum Zeichnen von Kurven auch eine nicht alltägliche Funktion zur Verfügung, und zwar wird dabei eine Kurve nach dem SPLINE-Verfahren erzeugt. Dazu können bis zu 20 Punkte beliebig auf dem Bildschirm plaziert werden, um die nach mathematischen Gesichtspunkten eine

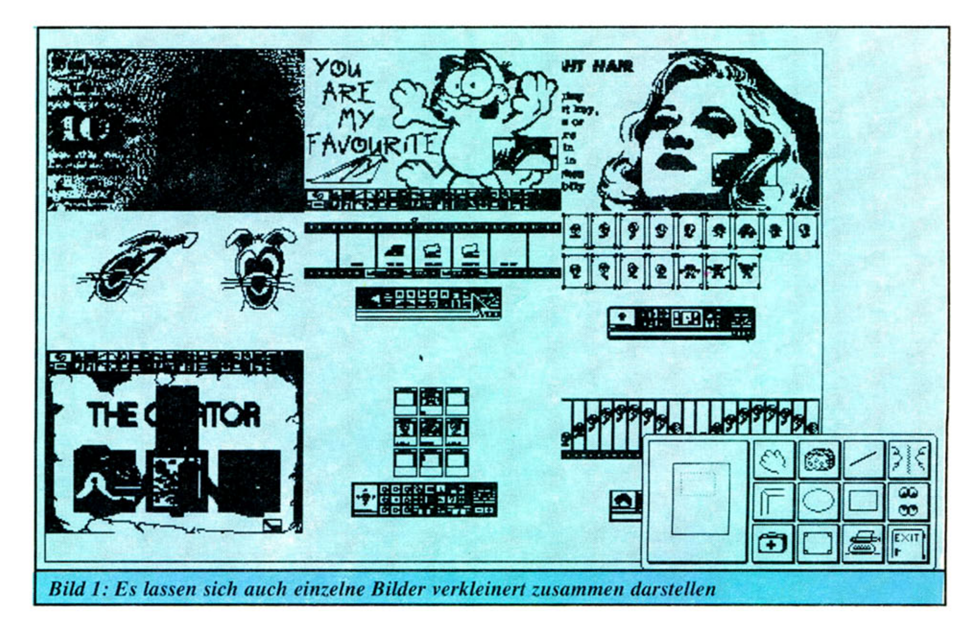


Bild 1: Es lassen sich auch einzelne Bilder verkleinert zusammen darstellen

Kurve konstruiert wird. Da der Verlauf im voraus aber nur schwer bestimmt werden kann, ist die nachträgliche Verschiebung aller Punkte zur Korrektur möglich. Die Darstellung der Linie erfolgt dabei in aufregender Geschwindigkeit (wie eigentlich alles), und es macht einfach Spaß, Kurven zu erzeugen.

Beim Zeichnen von Kreisen, Ellipsen, Rechtecken u.ä. kann bestimmt werden, ob diese vom Mittelpunkt oder in einer nachträglich veränderbaren Box aufgezogen werden sollen. Ferner lassen sich fünf verschiedene Bildpunkte abspeichern, die zum genauen Positionieren von Kreisen etc. dienen, falls man mehrere umeinander vom selben Mittelpunkt aus aufziehen will.

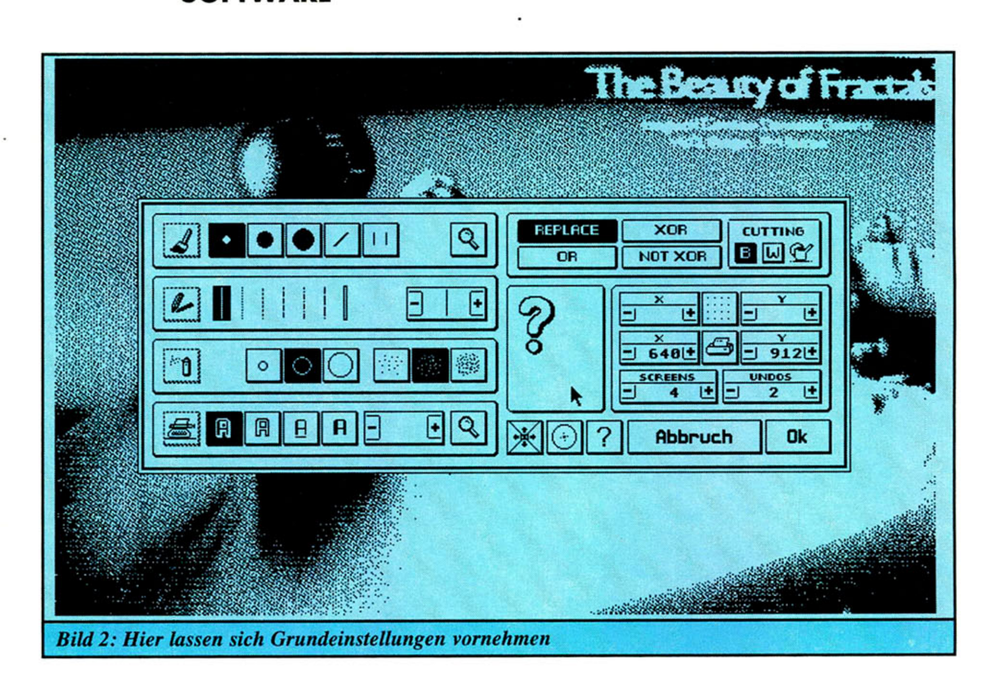


Bild 2: Hier lassen sich Grundeinstellungen vornehmen

Auf Biegen und Brechen

Auch auf eine ganze Reihe von Verfremdungseffekten kann der Creator-Besitzer zurückgreifen. Grundfunktionen sind hier Spiegeln (horizontal, vertikal), Biegen und Knicken. Leider hat er keine frei definierbare Spiegelachse, was die ganze Sache perfektionieren würde. Darüber hinaus kann man ein Objekt gradweise rotieren lassen. Besonderes gibt es auch wieder beim Vergrößern und Verkleinern. Hier wurde ein bißchen beim Macintosh abgeguckt (was ja nie schaden kann!). Durch Drücken der Shift-Taste ist eine maßstabsgerechte Vergrößerung bzw. Verkleinerung möglich. Genauso läßt sich beim Verziehen durch Tastendruck wahlweise die Höhe oder Breite eines Objektes fixieren. Auf den Funktionstasten kann man fünf verschiedene Vergrößerungsfaktoren speichern und sie so per Tastendruck wieder abrufen. Besondere Verzerreffekte erreicht man mit der Trapezfunktion, bei der ein Objekt perspektivisch verzerrt werden kann.



Bild 3: Die Bedienungsleiste zeigt nur ein Bruchteil der möglichen Funktionen

Reden ist Silber, Schreiben ist Gold

Zum Zeichenteil ist wohl auch der integrierte Texteditor zu rechnen, den man aber nicht mit einem normalen Editor vergleichen kann. Er hat ja auch eine andere Zielsetzung. Bevor man mit dem Editor arbeiten kann, sind zwei Arbeitsschritte nötig. Zum einen muß ein Zeichensatz (z.B. einer der weitverbreiteten von Signum!) geladen werden (s. auch Tabelle 2), zum anderen ist die Plazierung einer Box, in die der Text fließen soll, nötig. Nun kann der Text eingegeben oder

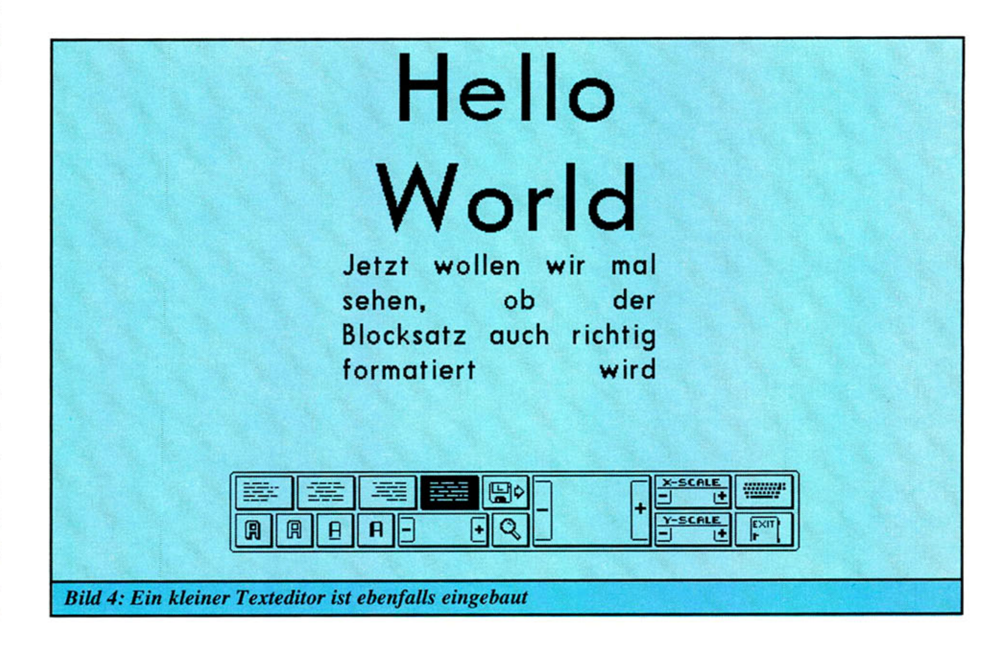
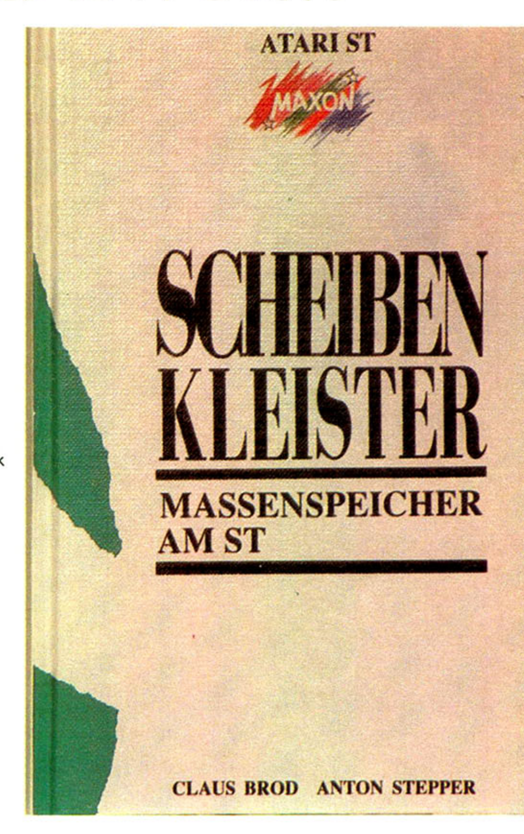


Bild 4: Ein kleiner Texteditor ist ebenfalls eingebaut

Aller guten Dinge sind zwei...



SCHEIBENKLEISTER
Massenspeicher am ST

Alles über Floppies, Festplatten und andere Massenspeicher am ST von Claus Brod und Anton Stepper.

Kursteil (für die ganze Familie):
- Floppyprogrammierung mit allen erlaubten und unerlaubten Mitteln (per BIOS, XBIOS, GEMDOS und direkter Controllerprogrammierung)
- Kopierschutz, Aufzeichnungsverfahren, Datenstrukturen auf der Diskette
- Hardwaredokumentation zu Floppy und Festplatte (Anschluß von Fremdlaufwerken, Justierung, Reparatur)
- Festplatte: Prinzip, Controller, Programmierung

Nachschlageteil (für Programmierer):
- Hard- und Softwarereferenz zu DMA-Chip, Floppycontroller, Festplattencontroller
- GEMDOS-, BIOS- und XBIOS- Funktionen zur Massenspeicherprogrammierung (auch als GFA-BASIC- Bibliothek auf Diskette)
- Systemvariablen (auch bisher undokumentierte), physikalische Grundlagen, Pinbelegungen und Ports.

Software (für alle, fertig zum Anwenden mit Anleitungen):
- **TED**, der Trackeditor: Formate analysieren, ändern, erstellen; Zugriff auf alle Controllerfunktionen
- **SED**, der Datei- und Sektormonitor für **RAM-Disk**, **EPROM - Disk**, Floppies und Festplatten: Ordnernamen ändern, gelöschte Dateien retten, spezieller Harddiskmonitor für direkten Festplattenzugriff (eigene Formatierroutine für max. drei MB mehr)
- neue HYPERFORMAT-Version 3.0: Bis zu 950 KB auf doppelseitiger Diskette, superfixe Formatierroutinen (optional unter 20 Sekunden für doppelseitige Disketten), Schnelladeformate
- Steprateneinstellung, Konvertierung von Disketten auf Schnelladeformat
- Assemblerroutinen für direkten Floppy- und Festplattenzugriff zum Einbinden in eigene Programme
- lauffähige Programme mit Quelltext auf Diskette.

582 Seiten, Buch mit Diskette für DM 59.-



OMIKRON.BASIC 3.0
Das Buch zum Handbuch

Alles über das neue ATARI-BASIC von Clemens Hoffmann

Einführung (für den absoluten Neuling):
- Kurze Anleitung in der BASIC-Programmierung.
- Über das Handbuch hinausgehende Beschreibung vieler Befehle, Besonderheiten und Kniffe. Verwendung selbstdefinierter Prozeduren und Funktionen. Viele Beispiele, Aufgaben mit Lösungen.

Für den Aufsteiger, aber auch für den geneigten Anfänger:
- OMIKRON.**Sprites** -Tücken, Vorteile, Anwendung.
- **Overlay**-Technik (Auslagern langer Programmteile und Laden bei Gebrauch).
- Grundlagen der strukturierten Programmierung. Schreiben eigener und Verwenden fremder Libraries (Bibliotheks-Funktionen).
- Aufrufe von TOS und GEM im BASIC (**GEMLib**); endlich die Wahrheit über die GEM-Aufrufe! Dabei wird auch das GEM-Zusatzprogramm **GDOS** berücksichtigt.
- **Sound** und **Grafik**-Programmierung. Grafische Effekte (z.B. die Verwendung mehrerer Grafik-Bildschirme und Zeichnen in nicht sichtbare Bildschirme. Aufbau von Metafiles, IMG-Bildern u.a.
- Aufbau und Verwenden der Menü-Leisten in **GEM-Accessories** in OMIKRON.BASIC.
- Verwendung der BASIC-internen **Multitasking**-Befehle.
- Einige Libraries (Turtle-Grafik Erweiterungen und Korrekturen zur GEMLib, usw).
- Erklärung der Befehle der Version 3.0 und ihre Anwendung. Die Feinheiten des Compilers V 2.0.
- Natürlich befinden sich **alle** Programme und Beispiele auf Diskette!

355 Seiten, Buch mit Diskette für DM 59.-



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ______________________

Vorname: ______________________

Straße: ______________________

Ort: ______________________

Unterschrift: ______________________

Hiermit bestelle ich:

☐ Exemplar "Scheibenkleister, Massenspeicher am ST" mit Diskette für DM 59,00

☐ Exemplar "OMIKRON.BASIC 3.0" mit Diskette für DM 59,00

Versandkosten: DM 7,50
Nachnahme zzgl. DM 3,50 Nachnahmegebühr.

☐ Vorauskasse
☐ Nachnahme

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrtieb in Holland: JOTKA COMPUTING Postbus 8183 NL-6710 AB Ede Tel.: 08380/38731 Fax.: 08380/21675

Extension	Art
E24	SIGNUM!-Editor-Zeichensatz
P09	SIGNUM!-9 Nadel-Drucker-Zeichensatz
P24	SIGNUM!-24 Nadel-Drucker-Zeichensatz
L30	SIGNUM!-Laserdrucker-Zeichensatz
FNT	GEM-Zeichensätze (STAD, System, Degas)
SFT, DAT	MonoSTar plus-Zeichensatz
CFN	Creator-Zeichensatz

Tabelle 2: In Creator verwendbare Zeichensätze

sinnvoll für längere Textpassagen auch als ASCII-Text von Diskette geladen werden. Unschön ist nur, daß bei zu langem Text der Rest, der nicht mehr in die Box paßt, einfach abgeschnitten wird. Man sollte also immer darauf achten, daß die Textbox ausreichend groß ist. Die Formatierung (in Echtzeit) kann auf vier Arten eingestellt werden. Links-, rechtsbündig, zentriert und Blocksatz stehen zur Auswahl.

Neben verschiedenen Schriftattributen (fett, unterstrichen, hohl usw.) ist die Schrägstellung von Zeichen in beliebigem Winkel möglich, oder einzelne Zeichen können mit dem integrierten Fonteditor jederzeit verändert werden. Man kann die Fähigkeiten des Creators in diesem Punkt zwar nicht mit denen eines DTP-Programm vergleichen, aber für kleinere Aufgaben der Seitengestaltungen mit Text und Grafik reicht es aus.

Creator hat Format

Das Bildschirmformat ist normalerweise auf die bestmögliche Ausgabemöglichkeit abgestimmt. Deshalb sollte man sich bei der Einstellung der Bildauflösung immer an der Auflösung des Druckers und nicht der bestehenden Bilder orientieren. Aus diesem Grund wird bei der Einstellung der Bildparameter zunächst der Druckertyp verlangt. Creator unterstützt hier 9 Nadel-, 24 Nadel- und Laserdrukker. Aus dieser Angabe wird dann automatisch zwischen drei verschiedenen Punktdichten (Maßeinheit immer DPI) ausgewählt. Als feste Formatgrößen stehen DIN A6 quer und hoch, DIN A5 quer und hoch und DIN A4 hoch zur Verfügung. Man kann die Bildauflösung natürlich auch frei einstellen, sollte dann aber darauf achten, daß sie auf keinen Fall größer als die Druckerauflösung ist.

Für nicht ganz so begabte Zeichner, oder wenn es auf Realitätsnähe ankommt, können Bilder mittels Scanner eingelesen werden. Zur Zeit werden der Hawk CP-14 sowie der Handy Scanner unterstützt. Geeignete Vorlagen vorausgesetzt, und mit etwas Nachbehandlung sind hervorragende Ergebnisse erzielbar, die natürlich auch den Weg aufs Papier finden.

Die Schere gewetzt

Als Blockoperationen stehen zunächst einmal normales Ausschneiden, ein Lasso und eine Schere zur Verfügung. Das Objekt läßt sich nach dem Ausschneiden auch als Pinsel einsetzen. Ferner lassen sich Mehrfachkopien anfertigen, bei denen ein ausgeschnittener Bildteil sooft wie möglich auf eine vorher definierte Seite kopiert wird. Objekte lassen sich ebenfalls mit einem Schatten oder einem Verlauf versehen oder sich "outlined" (ausgehöhlt) oder verdickt darstellen. Wer's gerne verziert mag, kann einen Zierrahmen um seine Zeichnung ziehen. Ab Version 1.1 wird sogar ein eigener Rahmeneditor für diese Funktion zur Verfügung stehen.

Alle ausgeschnittenen Objekte werden auf einem Klemmbrett (neudeutsch Clipboard) abgelegt. Dieses Clipboard faßt bis zu 36 Objekte beliebiger Größe, die sich alle auf einmal anzeigen lassen (gegebenfalls in verkleinerter Darstellung). Es bildet die Schnittstelle zum zweiten Teil des Programmes, dem Animationsteil.

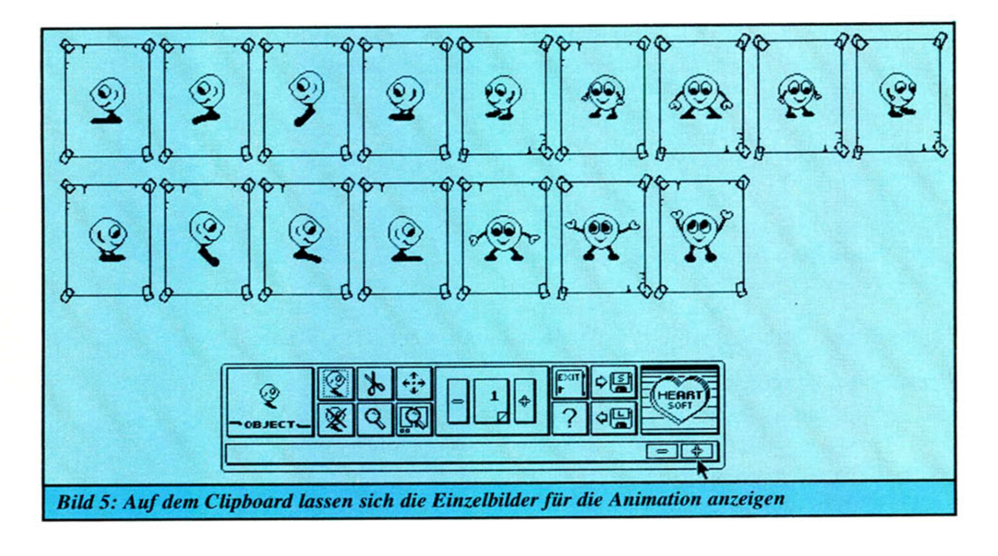


Bild 5: Auf dem Clipboard lassen sich die Einzelbilder für die Animation anzeigen

Vom Maler zum Filmemacher

Hat man im Zeichenteil die Grundvoraussetzungen dafür geschaffen, Animationen zu erstellen, kann man schnell feststellen, daß der Animationsteil noch umfangreicher ist. Obwohl sehr viel Wert auf eine leichte Benutzerführung gelegt wurde, ist das Erstellen einer halbwegs guten Animation schon ein sehr komplexer Vorgang. Man sollte sich deswegen immer erst einmal überlegen, was man erstellen will und ein kleines Storyboard schreiben. So erspart man sich eine Menge Arbeit und Änderungen.

Wir müssen zugeben, daß wir uns die Mühe des Selbstzeichnens erspart haben und eines der mitgelieferten Beispiele (eine kleine Raupe) benutzt haben, um einen Trickfilm zu erstellen. Wenn man bedenkt, daß eine Raupe sehr langsam ist, sie sich aber dennoch einigermaßen ruckfrei bewegen soll, werden schon mindestens 40 Einzelbilder benötigt, um sie von rechts nach links über den Monitor laufen zu lassen. Die Positionierung der Objekte für eine Szene geht zwar sehr einfach und recht schnell, aber 40 Bilder auf dem Monitor zu plazieren (ein paar mehr hätten auch nicht geschadet) benötigt doch einiges an Zeit, und dann hat man noch nicht einmal einen Hintergrund bzw. eine Umgebung, geschweige denn einen Vorspann, der sich zum Beispiel aus frei herumfliegenden Buchstaben vollautomatisch zusammensetzen könnte. Naja, für eine Raupe wäre das wohl auch zu viel des Guten. Sollten Sie aber genügend Zeit und ein wenig Geschick haben, wird es Ihnen wirklich auf sehr einfache Art ermöglicht, ganze Filme mit Auf-, Ab- und Überblenden der einzelnen Szenen zu erstellen. Im folgenden soll ganz kurz auf die Erstellung eines Films eingegangen werden.

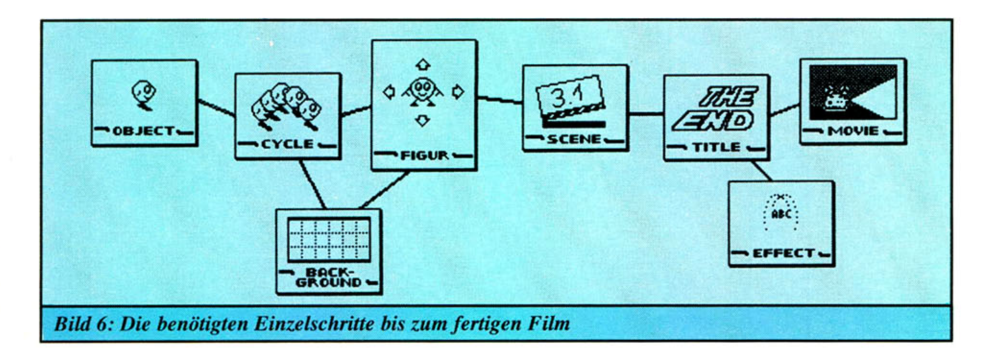


Bild 6: Die benötigten Einzelschritte bis zum fertigen Film

Schritt für Schritt

Im Animationsteil ist für jeden Teilschritt ein eigenes Bedienfeld vorhanden (es wird aber immer nur eines davon angezeigt). Die einzelnen Felder sind dabei sequentiell zu erreichen. Man kann also immer nur zwischen dem nächsten und dem vorherigen wählen. Einen Film erstellt man in mehreren Schritten. Zunächst erstellt man, wie bereits oben erwähnt, die einzelnen Objekte (das sind die Einzelbilder) und legt sie auf das Klemmbrett. Dann werden Objekte in Bewegung gesetzt. So entstehen verschiedene Zyklen, die man beliebig oft ablaufen lassen kann. Die Ablaufgeschwindigkeit eines Zyklus' ist frei wählbar.

Des weiteren können "Figuren" definiert werden. Eine Figur besteht aus verschiedenen, vorher zu definierenden Zyklen, welche die Bewegungsabläufe des "Charakters" in die acht möglichen Richtungen bestimmt. Dabei können sich die Bewegungen sogar dem Untergrund anpassen. Dazu stehen der Figur in jede Richtung neun verschiedene Bewegungsformen zur Verfügung. Es läßt sich sogar eine künstliche Schwerkraft für eine Figur anwählen. Eine Figur im Creator dient hauptsächlich der Koordinierung von Bewegungszyklen, die auch sofort in einem Direktsteuermodus durch Maus oder Tastatur auf korrektes und flüssiges Zusammenspiel der einzelnen Zyklen hin untersucht werden kann.

Ein Film besteht wie ein Tonband aus mehreren Spuren, die alle einzeln erstellt werden und sich dann im Film überlagern. Diese Spuren enthalten alle Objekte des Filmes und lassen sich beliebig in ihrer Darstellungsebene definieren. Creator verwaltet insgesamt 32 verschiedene Spuren. Bewegt sich ein Objekt in einer der hinteren Ebenen, kommt es natürlich vor, daß es von weiter vorne liegenden Objekten verdeckt wird. So sind sehr realistische, räumliche Effekte zu erzielen. Zusätzlich läßt sich noch ein beliebiger Hintergrund, z.B. eine Landschaft, positionieren, über die dann die einzelnen Objekte bewegt werden können.

Sind solche Szenen (max. 99) für den Film soweit fertig, geht es ans "Schneiden". Für den Creatoranwender heißt das, er kann die erstellten Szenen beliebig aneinanderreihen und zwischen die Szenen verschiedene Blendfunktionen legen. Davon bietet der Creator einige an wie z.B. Auf- und Abblenden. Aber auch die anderen sind nicht schlecht, z.B. kann der neue Szenenhintergrund von unten in den Bildschirm geklappt werden. Den besten Eindruck würde natürlich eine Demonstration der bewegten Bilder vermitteln, dies ist aber leider in einer Zeitschrift nicht möglich. Die Funktionsweise muß man einfach gesehen haben.

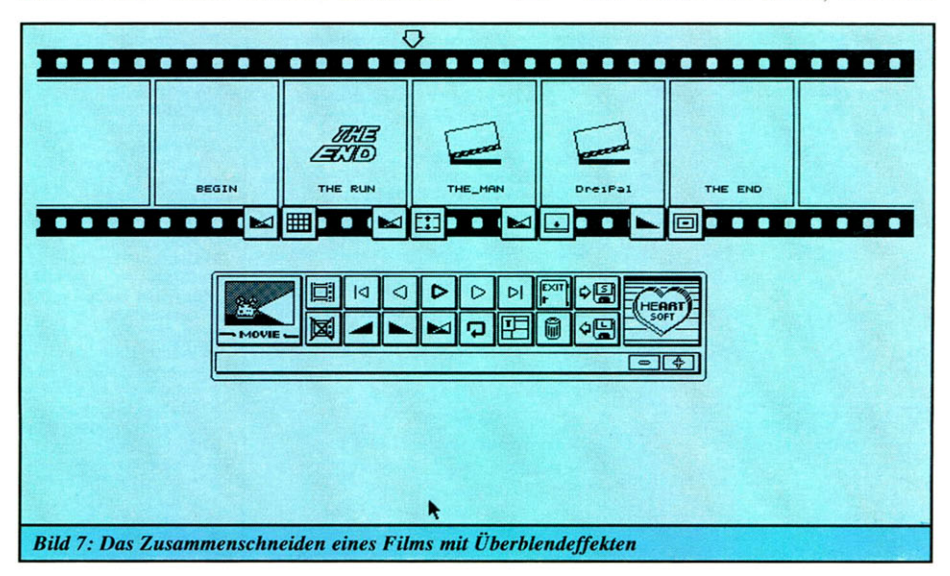


Bild 7: Das Zusammenschneiden eines Films mit Überblendeffekten

Ein einfaches Intro kann man z.B. erreichen, indem man einen Text im Zeichenteil schreibt und anschließend im Animationsteil für jeden einzelnen Buchstaben einen Bewegungsablauf mittels frei verschiebbarer Bahnen definiert und am Ende zusammentreffen läßt, wodurch der eigentliche Text entsteht. Viele solcher Abläufe sind bereits fertig auf Diskette vorhanden und brauchen nur noch geladen zu werden. Natürlich lassen sich bestehende Abläufe auch nachträglich noch verändern.

Zum Abschluß muß man nur noch die einzelnen Szenen und Titeleffekte zusammensetzen und erhält den fertigen Film. Natürlich lassen sich alle Teilschritte abspeichern und laden.

Fazit

Allein mit dem Zeichenteil des Creators dürften schon neue Maßstäbe bei den Zeichenprogrammen in dieser Preisklasse gesetzt werden. Natürlich bieten andere Programme auch wieder andere Zusatzfunktionen wie Musikeinbindung oder 3D-Teil, so daß man für sich selbst ganz individuell entscheiden muß. Wer allerdings gerne Trickfilme und Animationen erstellt, dem kann an und für sich nichts anderes empfohlen werden. Viele dürften aber von der Komplexität des Programmes zunächst überfordert sein, denn um die Feinheiten richtig einsetzen zu können, bedarf es eines gründlichen Studiums des Handbuchs und einiger Übung. Unübersehbar ist die Anlehnung an den Macintosh, wodurch aber zwangsweise der GEM-Standard fehlt. Das Benutzen von Accessories wäre wünschenswert - man denke nur daran, daß man mühsam eine Szene erstellt und keine formatierte Diskette mehr zur Vefügung hat. Aber es war ja schon von einer Version 1.1 die Rede, in der dann vielleicht auch eine Formatieroption eingebaut werden kann. Laut Application Systems ist der geplante Erscheinungstermin Januar 1989. Wer also noch DM 249,- vom Weihnachtsfest übrig hat, dem sei der Creator wärmstens empfohlen.

WS/HE

Application Systems /// Heidelberg
Englerstr. 3
6900 Heidelberg
Tel.: 06221/300002

NEU & AKTUELL



Die neuartige Einführung in die Textverarbeitung, die verspricht ein Klassiker zu werden. Ein Buch, das man auch einmal am Wochenende lesen möchte – begeisternd, und manchmal auch bissig – über ein „Programm", das nie ganz ausgereizt wurde.

MERKMALE:

- Voll von Information, die menschlich verpackt ist
- Zeigt minutiös die Technik der Textverarbeitung, ohne ihre Faszination zu vergessen
- Vom Praktiker für Praktiker: Für Autoren und Ärzte... Sekretärinnen und Schwiegermütter... Journalisten und Geschäftsleute
- Für Anfänger, die schnell lernen wollen und nichts vom Computer verstehen
- Für Fortgeschrittene, die sich darüber hinaus für RAM-Disk, Zeit sparen, Grafik und Drucklegung interessieren
- Für Kenner, die mal sehen wollen, ob es bei der Textverarbeitung noch etwas gibt, das sie nicht wissen
- Mit vielen Beispielen und Grafiken auf über 300 Seiten

INHALT:

- Wie man sofort Text auf den Bildschirm bringt
- Der unendliche Zeichensatz und 1st Proportional
- Alle Menüs auf einen Blick
- Tippen, Korrigieren, Zeitsparen, Wörterbuch...
- Formatieren und Umbruch
- Einbinden von Grafik, DEGAS im Text, selber malen, drucken
- Ist eine Art DTP möglich?
- 1ST MAIL – auf 30 Seiten so erklärt, daß es jeder versteht
- Zum ersten Mal: Vom Manuskript zum Buchdruck!
- Was man nicht im Handbuch findet: 100 Tips u. Tricks
- Verschiedene Bildgrößen
- Wie man den Treiber zum Großschreiben bringt
- „Was tun wenn...?" Eine lange Liste von Ratschlägen
- Der Umgang mit dem Drucker – Dichte, Schnelligkeit, Zeichensatz, Papiereinzug, Berechnung von Kopf- und Fußzeile
- Übersetzung der wichtigsten Fachausdrücke
- „1ST WORD TRAINER" – Aufgaben, Probleme, Fragen: Ein Übungsteil, der zum Denken auffordert.

Über 300 Seiten
B-414 **DM 49,-**



Neue erweiterte Auflage

MERKMALE:
Der Grundlehrgang ist das Buch für den richtigen Einstieg mit dem ATARI ST. Auf über 450 Seiten wird der Leser leicht verständlich in die Bedienung des Rechners eingeführt. Die vorliegende dritte Auflage wurde von Grund auf überarbeitet und stark erweitert. Sie berücksichtigt nun alle neuen Rechner (inclusive MEGA ST) und bietet eine gründliche Einführung in BASIC. Auf der beiliegenden Programmdiskette sind alle Beispielprogramme enthalten. Für Festplattenbesitzer ist eine Beschreibung der ATARI-Harddisk-Treibersoftware sowie ihres Einsatzes enthalten. Auf der Programmdiskette ist außerdem ein Back-up-Programm gespeichert.
Ein Utility-Kapitel beschreibt die auf der Diskette befindlichen Hilfsprogramme, die eine Grundausstattung jedes Software-Pools darstellen.
Professionelleren Ansprüchen genügen die Programme, die Ihnen in einem weiteren Abschnitt vorgestellt werden. Ihre Vor- (und Nach)teile erfahren Sie in kurzen Beschreibungen der Software.

INHALT:

- Entwicklungsgeschichte
- Die Hardware
- Das Betriebssystem
- Arbeiten mit der Festplatte
- ST-BASIC
- Nützliche Hilfsprogramme **auf Diskette**
- Vorstellung ausgewählter Programme
  - Signum!2
  - ST-Pascal Plus
  - Omikron-BASIC
  - GFA-BASIC
  - Tempus
- Anhänge
  - Index
  - Worterklärungen
  - Tabellen
  - Literaturverzeichnis

Hardcover, 453 Seiten
mit Programmdiskette
B-400 **DM 59,-**



MERKMALE:
Als optimale Ergänzung zum Handbuch bietet sich dieses Buch an. In zwei Hauptteilen wird zunächst eine systematische Einführung in die Programmierung von BASIC unter Berücksichtigung der besonderen Fähigkeiten des neuen GFA-BASICs gegeben.
Doch auch BASIC-erfahrene Programmierer lernen hier die neuen Schleifenstrukturen (es gibt noch mehr als FOR, WHILE und REPEAT) kennen.
Der zweite Teil baut auf dem ersten auf und vermittelt weitere Kenntnisse der Programmierung anhand von Programmen, die wiederum ausführlich beschrieben und erklärt sind.
Durch zahlreiche Anhänge, einen Index, sowie weiteren nützlichen Tabellen, wird das Buch optimal ergänzt.

INHALT:

Erklärung der Schleifen- und Programmstrukturen
- Primzahlenberechnung
- Zahlenraten

Variablentypen und Arrays
- Sieb des Eratosthenes
- Adreßeingabe

Unterprogramme und Prozeduren
- Rekursionen
- Labyrinthsuche

Multitasking in GFA-BASIC
Abstrakte Datentypen
- Druckerspooler
- Verkettete Listen
- Binäre Bäume

Sequentielle Dateiverwaltung
Random-Access-Dateien
- Verkettete Listen

Grafikprogrammierung
- Turtlegrafik
- Arbeiten mit mehreren Bildschirmen

Betriebssystemprogrammierung
- Aufrufen von TOS-Befehlen
- Verwenden des GEMs
- Menüverwaltung unter GFA-BASIC
- Arbeiten mit Dialogboxen

Beispielprogramme
- Fakturierung
- universelle Datenverwaltung
- etc.

Anhänge
- Index
- sonstige Anhänge

Für Einsteiger, Fortgeschrittene und Profis

Über 580 Seiten
mit Programmdiskette **DM 59,-**

Heim Verlag
ATARI ST
C AUF DEM ATARI ST
Heim Verlag

MERKMALE:

- C ist die zweite „Muttersprache" des ATARI ST: schnell, komfortabel, kompakt im Code.
- „C auf dem ATARI ST" ist für Anwender geeignet, die Erfahrungen mit anderen Programmiersprachen gemacht haben. Das Buch behandelt den vollen Sprachumfang von Standard-C und verweist auf BASIC und Pascal.
- Besonderer Wert wird auf die Anschaulichkeit und Genauigkeit der Darstellung gelegt. Alle Programmbeispiele sind getestet und direkt in den Text übernommen.
- In C lassen sich einfache Programme von außerordentlicher Übersichtlichkeit und Klarheit schreiben.
- Dieses Buch wird Ihnen eine solide Grundlage für die Programmierung in dieser eleganten Sprache legen. Die besonderen Fähigkeiten des ATARI ST werden dabei nicht zu kurz kommen.

Wollen Sie in C programmieren, dann müssen Sie dieses Buch lesen. Alle Programmbeispiele gibt es auf der Programmdiskette.

INHALT:

- Die Programmiersprache C
- Programmieren mit Editor und Compiler
- C-Compiler für ATARI: Digital Research, Lattice, Megamax
- Grundlegende Elemente eines C-Programms
- Variablentypen
- Felder und Vektoren
- Ausdrücke
- Zeiger
- Speicherklassen
- Bitfelder
- Varianten
- Aufzählungen
- Dateien
- Diskettenhandling
- Einbindung von Assemblerprogrammen
- Bildschirmgrafik in C
- Fehler in den C-Compilern
- Tools u. a.

Buch incl. Programm-Diskette

DM 59,-

Alle Preise sind unverbindlich empfohlene Verkaufspreise.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ______________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

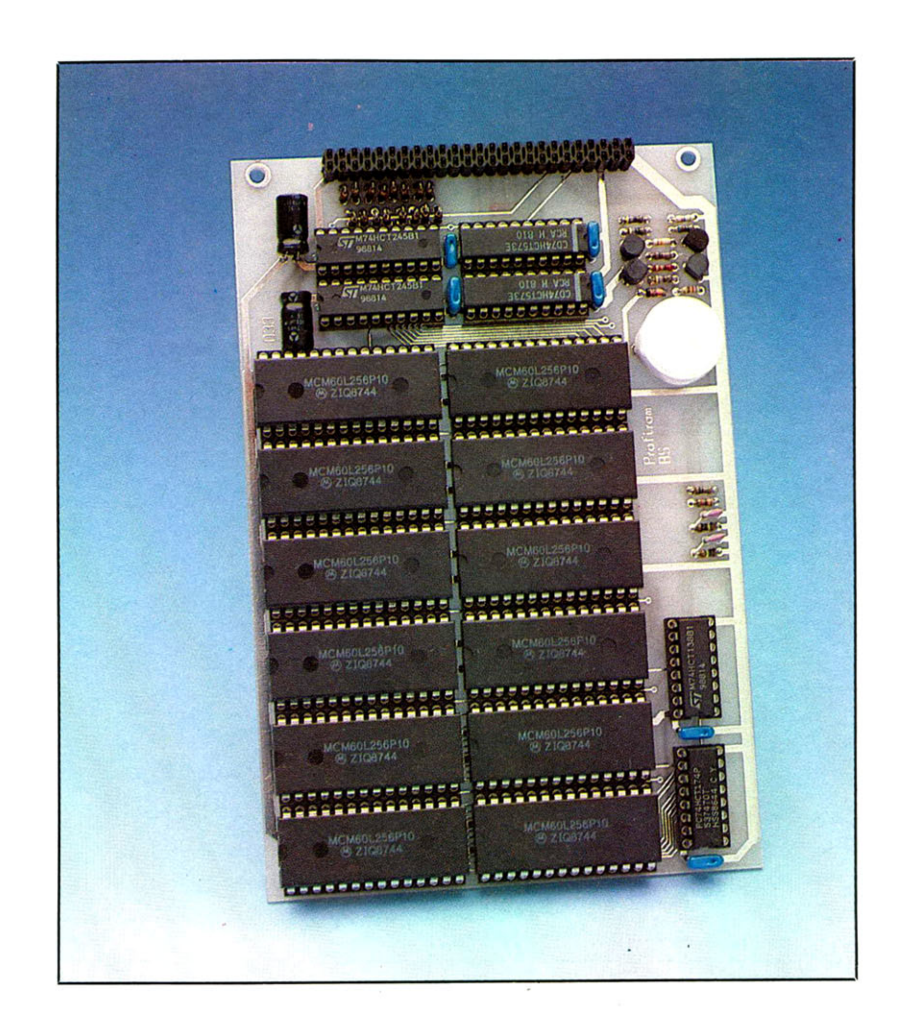
Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.



PROFIPORT

Anwendungen

Im vierten und vorerst letzten Teil wollen wir Ihnen ein paar Beispiele zur Anwendung des Profiports geben, wie z.B. Mehrkanal-Datenwandlung, Schalten und Steuern verschiedenster Lasten sowie die Ansteuerung von Schrittmotoren. Fangen wir also gleich damit an:



1. Netzverbraucher

1.1. Schalten von Netzverbrauchern

In Schaltbeispiel 1 ist die Möglichkeit aufgezeigt, mit einem *Opto-Triac-Netzverbraucher* zu schalten. Der Verbraucher sollte Überwiegend ohmisch sein, um eine Zerstörung des Foto-Triacs zu vermeiden. Störungen auf dem 220V-Netz werden durch den im Triac integrierten Nulldurchgangsdetektor auf ein Minimum reduziert. Ab 200 Watt Last sollte der Triac mit einem ausreichend großen Kühlkörper versehen werden. Dabei ist unbedingt zu beachten, daß die metallische Montagefläche des Triac Netzspannung führt (!) und deshalb isoliert montiert werden muß. Schlechthin sollten alle netzspannungsführenden Teile berührungssicher sein. Der Sicherheitsabstand zwischen netzspannungsführenden Teilen und dem Profiport muß größer als 8 mm sein.

An diesem Punkt soll ausdrücklich auf die Gefährlichkeit der Netzspannung hingewiesen werden. Falls Sie nicht die Erfahrung haben, eine derartige Schaltung in Betrieb zu nehmen, können Sie Ihr eigenes und das Leben anderer gefährden.

Die Ansteuerung des Netzschalters kann über jede Ausgangsleitung des Profiports erfolgen, wobei eine logische 0 an dieser den Verbraucher einschaltet.

1.2 Dimmen von Netzverbrauchern

Hier gilt im Prinzip das bereits Gesagte. Über ein 8 Bit-Datenwort wird der Anschnittwinkel und damit z.B. die Helligkeit eines Scheinwerfers von 0 bis 180 Grad bestimmt. Ist das Datenbyte Null, ist auch der Anschnittwinkel 0 und damit z.B. die Helligkeit des Scheinwerfers maximal. Zusätzlich bietet die Schaltung einen Ausgang, der für kurze Zeit während des Nulldurchgangs der Netzspannung aktiv wird. Das ist z.B. für netzsynchrones Auf- und Abblenden sinnvoll.

2. Schalten von Gleichstromlasten

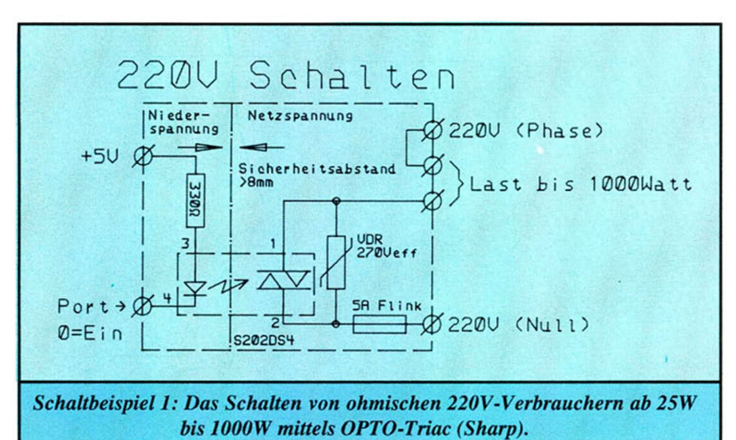
2.1. 8fach-Universaltreiber

Mit diesem Treiber ist jede Last, solange sie nicht mehr als 0,5A verbraucht, über den Profiport schaltbar. Die Sättigungsspannung der Treiber mit 1V begrenzt jedoch die Summe der Lastströme auf 1.5 A. Bei einer maximalen Schaltspannung von 50V können durch die integrierten Schutzdioden auch induktive Lasten geschaltet werden. Die Schaltzeiten liegen im Bereich einer Mikrosekunde.

2.2. Der Opto-Schalter

Wie schon gesagt, ist bei großen Lasten eine galvanische Trennung vom Profiport erstrebenswert. So auch beim Opto-Schalter. Er schaltet bis zu 4A bei 60V. Allerdings muß der Schalttransistor BD679 ab einer Last von 0,2A entsprechend gekühlt werden. Der Schalter begrenzt im Kurzschlußfall den Maximal-

strom auf einen über *RS* einstellbaren Wert. Die Verlustleistung (Sättigungsspannung mal Laststrom) darf in keinem Fall 40Watt übersteigen und muß über einen Kühlkörper abgeführt werden. Der Opto-Schalter eignet sich für Schaltfrequenzen bis zu 5KHz.



Schaltbeispiel 1: Das Schalten von ohmischen 220V-Verbrauchern ab 25W bis 1000W mittels OPTO-Triac (Sharp).

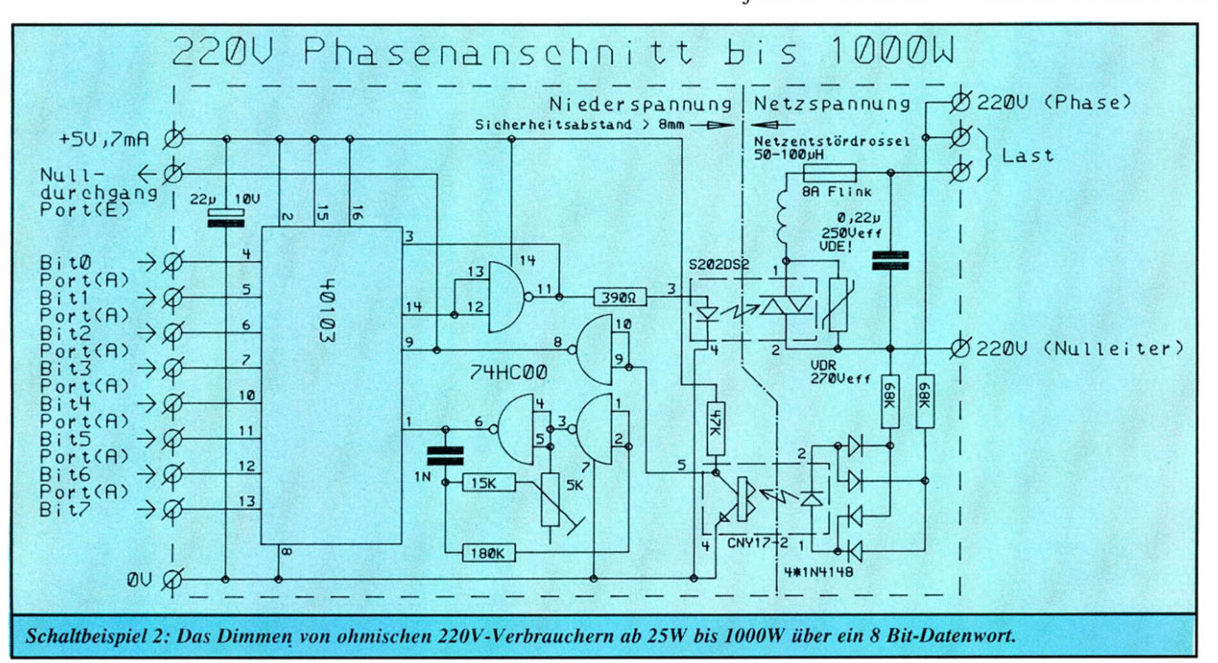
Auch hier ist der Schalter eingeschaltet, wenn der entsprechende Ausgang LOW ist.

2.3. Die Opto-Vollbrücke...

...ist primär zum galvanisch getrennten Steuern von Gleichstrommotoren gedacht. Dabei werden bis zu 4A bei Spannungen bis 60V an den Motor geliefert. Die vier Leistungstransistoren müssen ab 0.4A Laststrom gekühlt werden. Die Schaltung ist kurzschlußfest, jedoch müssen die beiden NPN-Transistoren BD679 im Kurzschlußfall die entstehende Verlustleistung auf sich nehmen, welche in jedem Fall unter 40W bleiben muß. Daher müssen diese eventuell einen größeren Kühlkörper erhalten. Der Kurzschlußstrom ist einstellbar über RS und errechnet sich aus 0,65V / RS.

3. Zwei-Phasen-Unipolar-Schrittmotorsteuerung

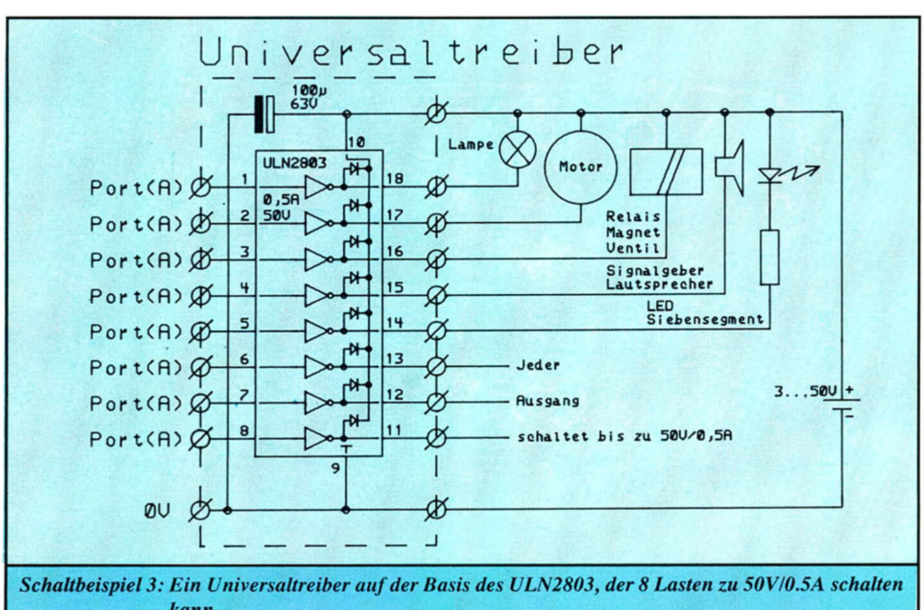
Diese Schaltung eignet sich im Speziellen für 2 Phasen-Unipolar-Schrittmotoren bei Strangströmen bis zu 0.5A und Betriebsspannungen bis 15 Volt. Passende Motoren sind reichlich in den Angeboten diverser Elektronikversände als Restposten zu finden. Damit der Schrittmotor sich dann auch bewegt, ist ein kleines Programm nötig, welches die im Bild gezeigten Steuerworte nacheinander an den Treiber legt.



Schaltbeispiel 2: Das Dimmen von ohmischen 220V-Verbrauchern ab 25W bis 1000W über ein 8 Bit-Datenwort.

3.1. Zwei-Phasen-Bipolar-Schrittmotorsteuerung

Hierbei handelt es sich um eine wesentlich effektivere Schaltung als oben. Sie ist in der Lage, bei einer Betriebsspannung bis 36V und Phasenströmen bis 2A etwa 80Watt an den Motor abzugeben. Außerdem arbeitet sie nach dem Prinzip der getakteten Konstantstromansteuerung, womit sich hinsichtlich Motordynamik (Beschleunigung) und Wirkungsgrad um Größenordnungen bessere Werte erreichen lassen. Jedoch ist es nicht sinnvoll, mit dieser Schaltung einen 'Spielzeugschrittmotor' anzusteuern, da dieser aufgrund seines hohen Innenwiderstandes nicht für Stromsteuerung geeignet sind. Weiterhin bietet die Schaltung eine Halbschrittoption, mit der sich die Schrittzahl pro Achsumdrehung erhöhen und damit Motorresonanzen reduzieren lassen.

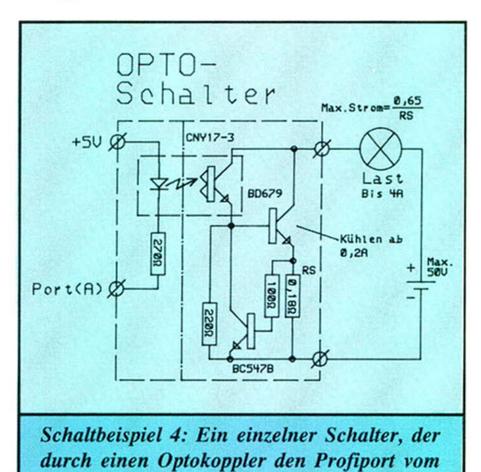


Schaltbeispiel 3: Ein Universaltreiber auf der Basis des ULN2803, der 8 Lasten zu 50V/0.5A schalten kann.

Die Zuleitungen einer Motorphase sollten, um Funkstörungen zu vermeiden,

verdrillt werden. Am besten eignen sich symmetrisch abgeschirmte Leitungen. Der L298 (SGS) muß außreichend gekühlt werden.

Die Steuereingänge sind galvanisch getrennt, um Störungen und Probleme zu vermeiden. Die abfallende Flanke am Eingang Schritt bringt den Motor über die am Eingang 'Richtung' bestimmte Richtung einen Schritt weiter.



Schaltbeispiel 4: Ein einzelner Schalter, der durch einen Optokoppler den Profiport vom Laststromkreis trennt.

Die Schrittfreqenz kann, je nach Motordaten, bis zu 10 Kilohertz betragen.

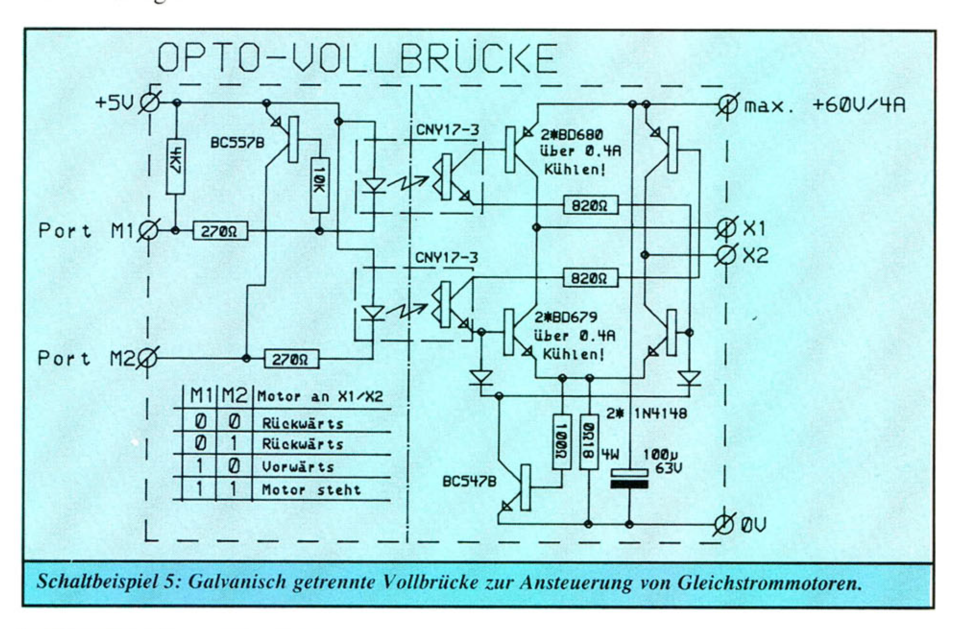
4. 8 Bit-Digital-/Analogwandler

Das 8 Bit-Eingangsdatenwort, multipliziert mit 0.01, ergibt die Ausgangsspannung in Volt. Sind alle Eingangsbits high, so sind es demnach 2.5 Volt. Der Ausgangswiderstand beträgt etwa 10 KΩ. Die Einschwingzeit der Analogspannung ist kleiner als 2 Microsekunden. LSB ist in diesem Fall das Bit mit der Wertigkeit Eins. Der Stromverbrauch aus der Profibank liegt um 10mA.

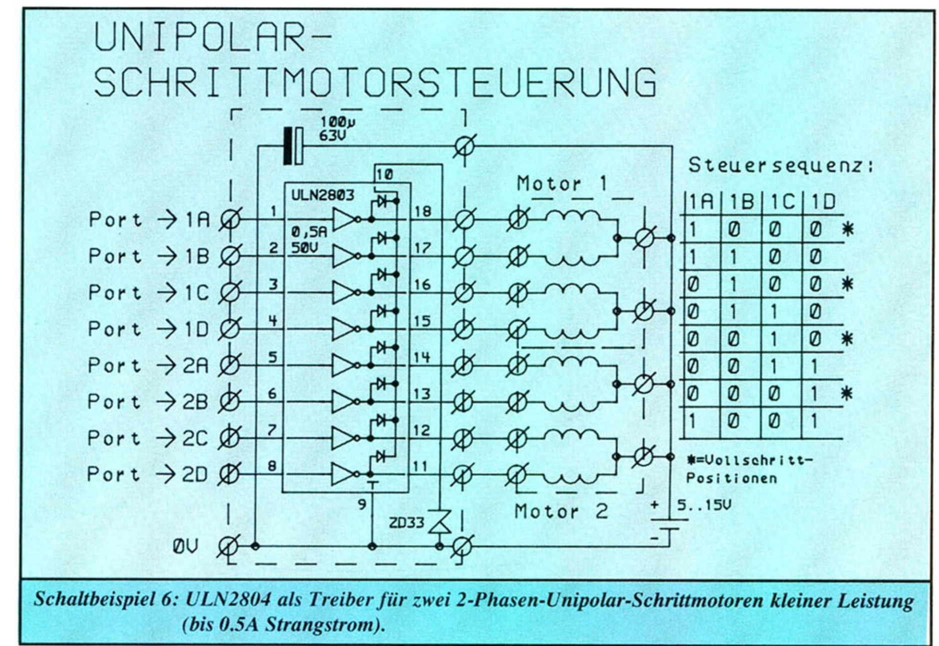
4.1 8 Bit - 11Kanal-Analog-/Digitalwandler

Diese Schaltung wandelt eine von 11 Analogspannungen in ein 8 Bit-Datenwort, das seriell übertragen wird. Der Wandler benötigt eine Referenzspannung zwischen 'Referenz-' und 'Referenz+', die 5.2 Volt nicht überschreiten darf. Zweckmäßigerweise legt man 'Referenz-' an 0V und die Referenzspannung zwischen 0V und 'Referenz+'. Je nachdem, wo der Analogwert zwischen den Referenzspannungen liegt, gibt der Wandler einen Wert zwischen 0 und 255 aus. Es sind 11 Eingänge vorhanden, wobei die Spannung an allen Eingängen (auch Referenzeingängen) die Betriebsspannungen nicht über- oder unterschreiten darf. Jeweils ein Eingang kann über die integrierte Abtast-Halte-Schaltung festgehalten und gewandelt werden. Das Programm zur Ansteuerung diese Wandlers finden Sie in Listing 1.

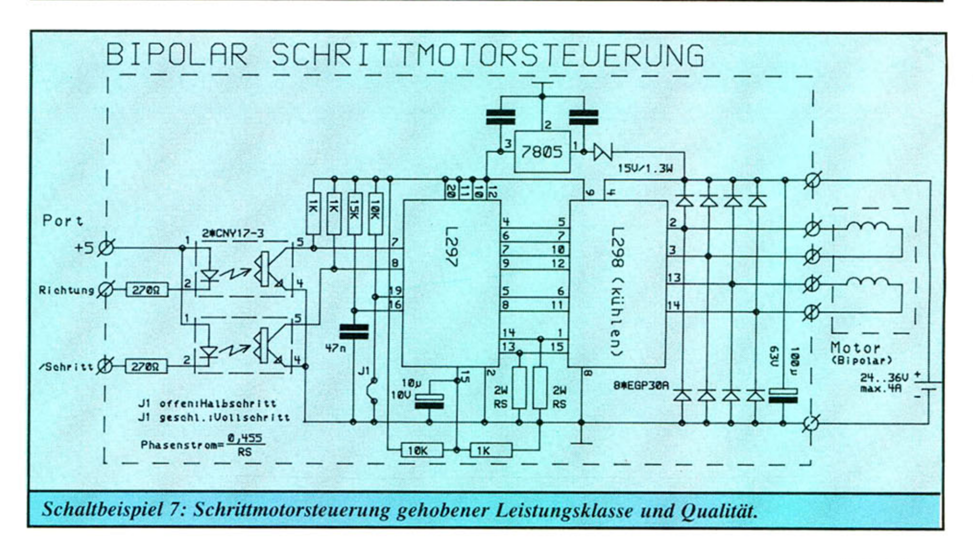
Alle Schaltungen sind lediglich Vorschläge und erheben keinen Anspruch auf Vollkommenheit. Fühlen Sie sich frei, diese Anregungen zu gebrauchen.



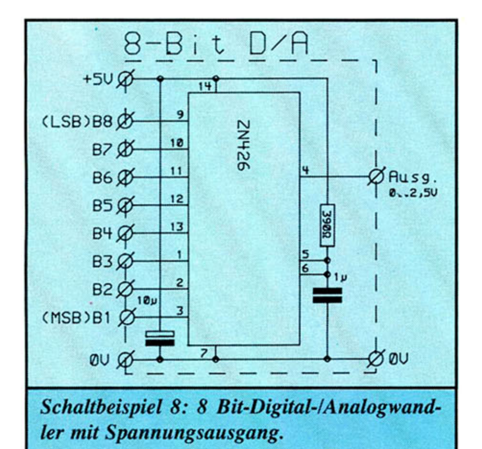
Schaltbeispiel 5: Galvanisch getrennte Vollbrücke zur Ansteuerung von Gleichstrommotoren.



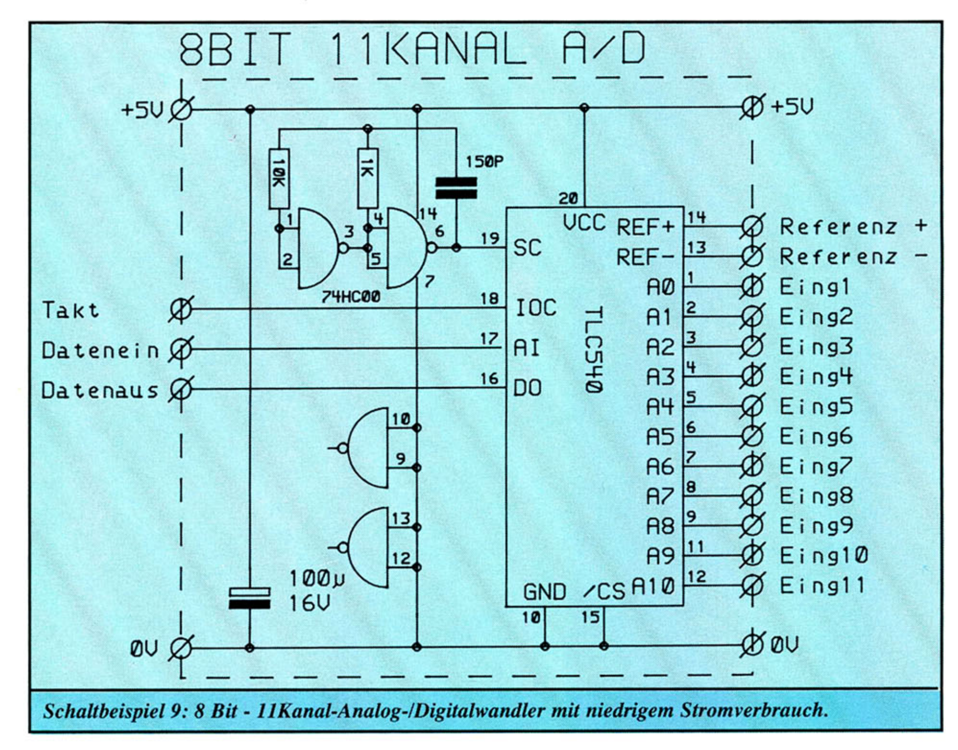
Schaltbeispiel 6: ULN2804 als Treiber für zwei 2-Phasen-Unipolar-Schrittmotoren kleiner Leistung (bis 0.5A Strangstrom).



Schaltbeispiel 7: Schrittmotorsteuerung gehobener Leistungsklasse und Qualität.



Schaltbeispiel 8: 8 Bit-Digital-/Analogwandler mit Spannungsausgang.



Schaltbeispiel 9: 8 Bit - 11Kanal-Analog-/Digitalwandler mit niedrigem Stromverbrauch.

```
 1:  ' ADU.BAS - Beispielroutine zur Ansteuerung des
                 11-Kanal-ADU am Profiport
 2:  ' R.Hofmann '88
 3:  ' Aufruf:
 4:  '
 5:  ' @Adu (Adr%,T%,De%,Da%,Ka%,*wert%)
 6:  '
 7:  ' Adr% - die Adresse des Portregisters, an dem
                           der ADU angeschlossen ist.
 8:  ' T%   - Bit dieses Ports, das mit der Taktlei-
                       tung des ADU verbunden ist.
 9:  ' De%  - dito, jedoch das Bit 'Datenein'
10:  ' Da%  - dito, jedoch das Bit 'Datenaus'
11:  ' Ka%  - Kanalnummer für den ADU von 0...10
12:  ' *Wert% - Der Übergabepointer für die
                Ergebnisvariable (Wert von 0...255)
13:  '
14:  Sdr%=&HFB0000! Basisadresse d.Schreibdatenreg.
15:  P1%=&HFAA000 ! Basisadresse der 1. PIA
16:  P2%=&HFAB000 ! Basisadresse der 2. PIA
17:  Crb%=0       ! Offset in das Kontrollregister B
18:  Cra%=2       ! Offset in das Kontrollregister A
19:  Prb%=4       ! Offset in das Daten- oder
                    Datenrichtungsregister B
20:  Pra%=6       ! Offset in das Daten- oder
                    Datenrichtungsregister A
21:  Lo%=&H800    ! Offset in den 'Lesen'-Ber. der PIA
22:  '
23:  ' Kurze Schleife, d.alle Kanäle a.d.Bildschirm
                                              zeigt
24:  Erg%=0
25:  Do
26:    Print At(0,0);
27:    For T%=1 To 10
28:      @Adu(P1%+Pra%,1,2,4,T%,*Erg%)
29:      Print "Kanal: ";T%;" : ";Erg%;"   "
30:    Next T%
31:  Loop
32:  '
33:  ' ********** Unterprogramm *********
34:  '
35:  Procedure Adu(Adr%,T%,De%,Da%,Ka%,Wert%)
36:    Local Bi%,B%,A%,We%
37:    We%=0
38:    A%=Dpeek(Adr%+Lo%)/256
39:    A%=A% And ((De% Or Ka%) Xor 255) ! Löschen des
                                      Takt- und Datenbits
40:    Void Peek(Sdr%+256*A%)
41:    Void Dpeek(Adr%)
42:    For Bi%=0 To 7
43:      Mul We%,2
44:      B%=A% Or De%*(Ka% And 8)/8
45:      Mul Ka%,2
46:      We%=We%+(Dpeek(Adr%+Lo%)/256 And Da%)/Da%
                                          ! Datenwort
47:      Void Peek(Sdr%+256*B%)
48:      Void Dpeek(Adr%)
49:      Void Peek(Sdr+256*(B% Or T%))
50:      Void Dpeek(Adr%)
51:    Next Bi%
52:    *Wert%=We%     ! Übergabe des Wandlerergeb.
53:    Void Peek(Sdr%+256*A%)
54:    Void Dpeek(Adr%)
55:  Return
```

ENDE

LineA für Turbo-C

Die meisten C-Compiler für den ATARI ST werden mit umfangreichen Bibliotheken ausgeliefert, welche die Programmentwicklung wesentlich erleichtern. Jedoch fehlt bei den meisten Bibliotheken die Möglichkeit, die LineA-Funktionen aufzurufen. Für den Programmierer heißt das, will er beispielsweise einen Punkt auf dem Bildschirm setzen, eine eigene Routine zu programmieren oder sich dem langsamen und umständlichen VDI (Applikation anmelden, Workstation öffnen, VDI-Aufruf, Workstation schließen, Applikation abmelden) anzuvertrauen.

Das hier vorliegende Programm soll da Abhilfe schaffen. Es ermöglicht den Aufruf der LineA-Funktionen von Turbo-C aus. Das Programm läßt sich mit geringem Aufwand auch an jeden anderen C-Compiler anpassen.

Installation der LineA-Bibliothek

- Geben Sie die Definitions-Datei ein und speichern Sie diese im Ordner *'INCLUDE'* unter dem Namen *'linea.h'*.
- Geben Sie das Assemblerlisting ein und lassen Sie vom Assembler eine Objektdatei erstellen (DR-kompatibel). Das Assemblerlisting ist mit dem GFA-Assembler 1.1 entwickelt worden, dürfte aber auch auf allen anderen Assemblern laufen, die DR-Objekt - code erzeugen können.
- Speichern Sie das Objektcode-File unter dem Namen *'linea.o'* im Ordner *'LIB'* ab.
- Erstellen Sie eine Projektdatei entsprechend Listing 2.

Wie funktioniert's?

Die Funktionsweise ist bei allen LineA-Aufrufen prinzipiell gleich und vollzieht sich in drei Schritten. Im ersten Schritt werden die Parameter in die LineA-Variablen eingetragen. Um eine höhere Geschwindigkeit zu erreichen, wird die Turbo-C Option der Registerübergabe ausgenutzt, d.h. die ersten drei Werte-Parameter werden in den Registern D0-D2, die ersten beiden Zeiger in A0-A1 übergeben. Die restlichen Parameter liegen ab *4(sp)* auf dem Stack (siehe Kapitel 7 des Turbo-C-Handbuchs). Im zweiten Schritt werden die Register D3-D7 und A2-A6 gerettet (Turbo-C hätte sie gerne unbeschädigt zurückerhalten). Schritt zwei kann auch vor Schritt eins ausgeführt werden, wenn keine Parameter auf dem Stack übergeben werden, da ein *'movem'*-Befehl den Stackpointer verändert (siehe Listing). Im dritten Schritt wird schließlich die entsprechende LineA-Funktion mit *'.dc.w $a00x'* aufgerufen, und daran anschließend werden die Register restauriert.

Die Anpassung an andere Compiler

Um das Programm mit anderen Compilern zu verwenden (nicht nur C-Compiler verwenden das DR-Format...), müssen Sie das Listing so ändern, daß alle Parameter vom Stack genommen werden. Wo auf dem Stack die Parameter liegen und welche Register gerettet werden müssen, entnehmen Sie bitte dem entsprechenden Handbuch. Beachten Sie bitte auch die Reihenfolge, in der die Parameter übergeben werden.

Die Benutzung in eigenen Programmen

Laden Sie in jedes Programm, das auf die LineA-Grafik zugreift, die Datei *'linea.h'* mit Hilfe der Preprozessor-Anweisung *'#include <linea.h>'*. Die besondere Art der Funktionsdeklaration von Turbo-C (siehe Kapitel 5 des Turbo-C-Handbuchs) ermöglicht es, daß grobe Fehler bei der Parameterübergabe vom Compiler erkannt und angezeigt werden. Achtung: Der Compiler erkennt nur Fehler bei unkorrekter Parameteranzahl und falschen Typen!!!

Beachten Sie bitte, daß die Compiler-Option Registerübergabe *eingeschaltet ist.*

Gehen Sie sehr sorgfältig bei der Nutzung von LineA vor. Unkorrekte Parameter können zu unerwünschten Resultaten oder gar zum Abstürzen des Betriebssystems führen!

Die Bedeutung der LineA-Parameter zu erklären, würde an dieser Stelle wohl den Rahmen sprengen. Die Bezeichnungen der Parameter entsprechen denen in der Fachliteratur. Einschlägige Bücher wie Data Beckers "ATARI ST Intern" oder das Markt&Technik-Buch "ATARI ST Assemblerbuch" geben jedoch knappe Auskunft über die Bedeutung der Parameter. Am besten probiert man nach dem Studium dieser Bücher ein wenig herum. Vielleicht schreibt ja auch mal jemand einem LineA-Kurs in dieser Zeitschrift...

Fehlersuche mit dem Debugger

Wenn sich ein Fehler bei der Benutzung der LineA-Bibliothek partout nicht finden lassen will, verwenden Sie einen Debugger zur Fehlersuche. Befreien Sie das fehlerhafte Programm mit den Befehlen '/*' und '*/' von unnötigem Ballast. Beim GFA-Debugger ist folgendes Vorgehen sinnvoll:

Laden Sie das Programm mit dem Befehl *' e "programmname" '* in den Debugger. Auf dem Bildschirm wird unter anderem die Startadresse des Text-Segments angegeben. Schalten Sie die Registerüberwachung mit *'regon'* ein. Tracen Sie nun das Programm mit dem Befehl *'ta Adresse_des_Textsegments'*. Anhand des gerade abgearbeiteten Befehls (die LineA -Routinen werden übrigens mit *'jsr $xxxxxx'* angesprungen) und der Registerinhalte sollten Sie nun den Fehler lokalisieren können. Um die LineA-Parameter zu überwachen, die auf dem Stack übergeben werden, erstellen Sie eine zweite LineA-Datei, bei der Sie hinter jeden *'move.x x(sp),x(a0)'*-Befehl die Anweisung *'move.x x(sp),d0'* setzen. In D0 können Sie sich so den Stack ansehen. Vergessen Sie nicht, vorher D0 zu retten. Nennen Sie die daraus entstehende Objekt-Datei *'lineadbg.o'* und erstellen Sie eine entsprechende Projektdatei. Wie Sie vielleicht bemerkt haben, sind nicht alle LineA-Routinen implementiert. Wenn Sie Lust haben, vervollständigen Sie die LineA-Bibliothek.

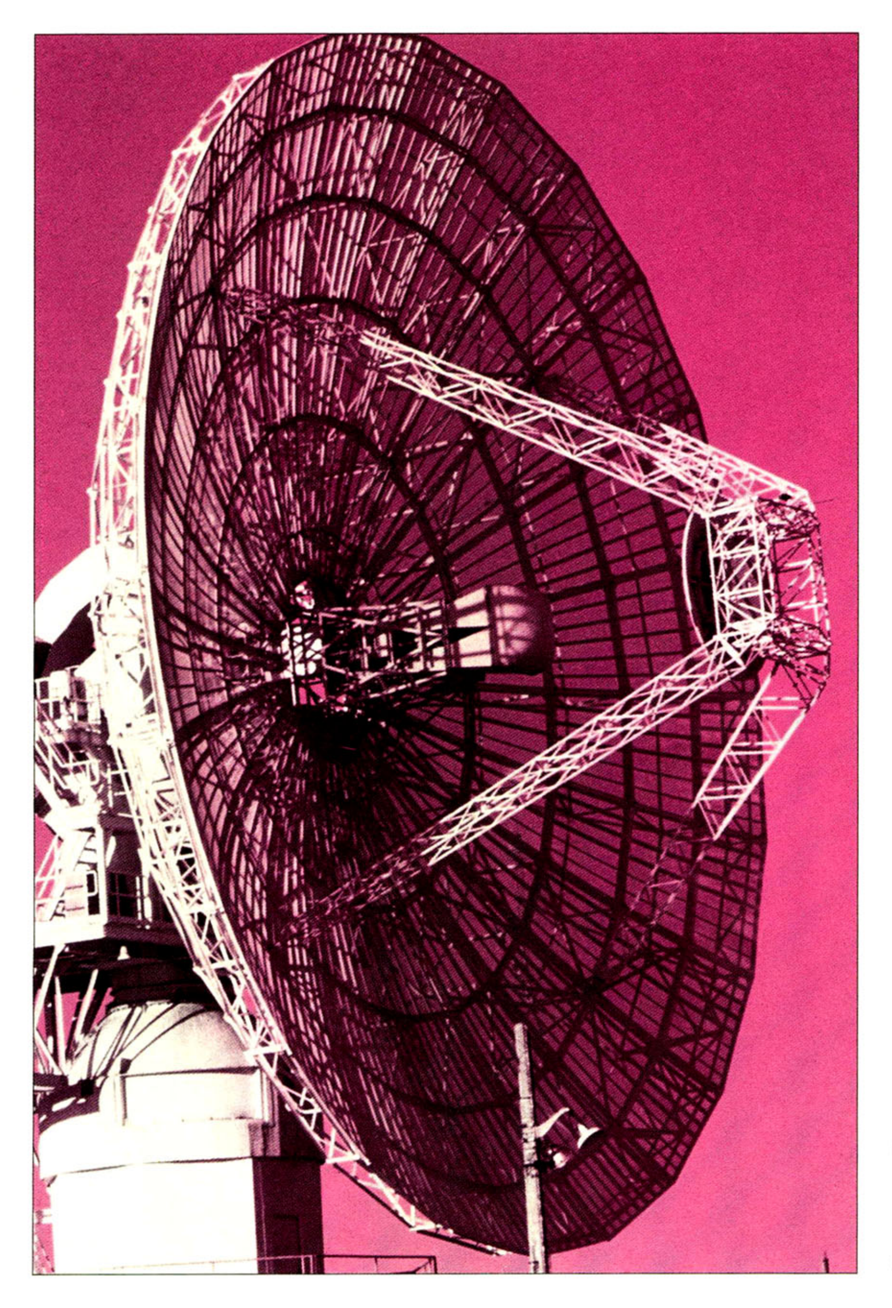
Thomas Schlesinger

```
 1:  /*
 2:
 3:         ********************************************************
 4:         *                                                      *
 5:         * Definitionsdatei für die LineA-Aufrufe unter Turbo-C *
 6:         *                                                      *
 7:         ********************************************************
 8:
 9:
10:                  geschrieben von Thomas Schlesinger
11:                               im  Oktober 1988
12:                               mit Turbo-C 1.0
13:
14:  */
15:
16:
17:  void Init_LineA     (void);
18:
19:  void Put_Pixel      ( int x, int y, int c );
20:
21:  int Get_Pixel       ( int x, int y );
22:
23:  void Line           ( int x1, int y1, int x2, int y2,
24:                        int fgbp0, int fgbp1, int fgbp 2,
25:                        int fgbp3, int lnmask, int wrtmod,
26:                        int lstlin );
27:
28:  void H_Line         ( int x1, int y1, int x2, int fgbp0,
29:                        int fgbp1, int fgbp2, int fgbp 3,
30:                        int wrtmod, int pattern[], intpatmask);
31:
32:  void Fill_Rect      ( int x1, int y1, int x2, int y2,
33:                        int fgbp0, int fgbp1, int fgbp 2,
34:                        int fgbp3, int wrtmod, int pattern[],
35:                        int patmask, int multfil, intclip,
36:                        int xminclp, int xmaxclp, intyminclp,
37:                        int ymaxclp);
38:
39:  void Fill_Poly      ( int ptsin[], int size, int y,
40:                        int fgbp0, int fgbp1, int fgbp 2,
41:                        int fgbp3, int wrtmod, int pattern[],
42:                        int patmask, int multifil, intclip,
43:                        int xminclp, int xmaxclp, intyminclp,
44:                        int ymaxclp);
45:
46:
47:  void Mouse_on       (void);
48:
49:  void Mouse_Off      (void);
50:
51:  void Mouse_Form     ( int mask[], int cursdt[] );
52:
53:  void Undraw_Sprite  ( int save[] );
54:
55:  void Draw_Sprite    ( int x, int y, int data[], int save[] );
56:
57:
58:  /* Hinweis:
59:
60:     Die Bedeutung der Parameter kann in einschlägige n Büchern,
61:     wie z.B. Data Becker's >Atari ST Intern< nachgelesen werden.
62:     Die Namen entsprechen den üblicherweise verw. Bezeichnungen.
63:     Bei den Funktionen Fill_Poly($a006), Fill_Rect($ a005) müssen
64:     die Clipping-Parameter mit übergeben werden. Falls nicht ge-
65:     clippt wird, übergibt man Scheinparameter(Dummies).
66:
67:  */
```

Listing 1: Header-Datei zur LineA-Bibliothek

```
 1:      ; DEFAULT.PRJ, angepasst an LineA-Bibliothek
 2:      ; ----------------------------------------
 3:  *   ; Programmname vom obersten Fenster
 4:  =              ;
 5:  tcstart.o      ; Initialisierungs-Modul
 6:  *              ;
 7:  linea.o        ; LineA-Bibliothek
 8:  tcfltlib.lib   ; Fliesskomma-Bibliothek
 9:  tcstdlib.lib   ; Standard-Bibliothek
10:  tcextlib.lib   ; Extended-Bibliothek
11:  tctoslib.lib   ; TOS-Bibliothek
12:  tcgemlib.lib   ; AES & VDI-Bibliothek
13:                 ; unnötige Bibliotheken entfernen
```

Listing 2: Neue Projektdatei



...UND PLÖTZLICH HAT IHR ATARI ST ODER MEGA ST EINE BEGEGNUNG DER BESONDEREN ART: VORTEX HD PLUS

Festplatten-Systeme von 20 bis 120 MB!

Das hat es bis jetzt noch nicht gegeben: Ein Festplatten-Programm für den ATARI ST bzw. MEGA ST mit Kapazitäten von 20 bis 120 MB formatierte Speicherkapazität!

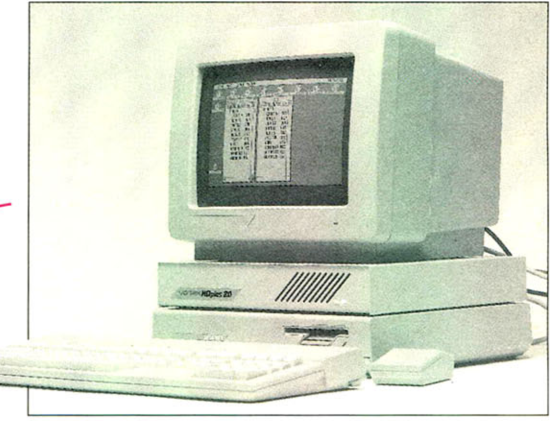


Weitere Vorteile

- Cache-Memory
- Auto-Parker
- bis zu 16 Partitionen
- Disketten-Backup-Programm
- bootfähig

Natürlich im ATARI-Design und in vortex-Qualität. Komplett anschlußfertig mit System-Diskette, Buskabel und deutschem Handbuch.

Holen Sie sich die kompletten Informationen. Sofort!





...UND PLÖTZLICH LEISTET IHR COMPUTER MEHR

I·N·F·O·S·C·H·E·C·K

Senden Sie mir umgehend alle Informationen über Ihr HDplus-Programm und nennen Sie mir den nächstgelegenen vortex-Vertragshändler.

vortex Computersysteme GmbH
Falterstraße 51–53 · 7101 Flein · Telefon (07131) 52061

```
1:   ;**********************************************
2:   ;*                                            *
3:   ;*   Line-A Grafik-Bibliothek für Turbo-C 1.0 *
4:   ;*                                            *
5:   ;*                                            *
6:   ;*   geschrieben von Thomas Schlesinger       *
7:   ;*              am  13.10.1988                *
8:   ;*              mit dem GFA-Assembler         *
9:   ;*                                            *
10:  ;**********************************************
11:
12:
13:
14:  ; ** Alle Funktionen dem Linker bekanntmachen **
15:
16:          .GLOBL Init_LineA
17:          .GLOBL Put_Pixel
18:          .GLOBL Get_Pixel
19:          .GLOBL Line
20:          .GLOBL H_Line
21:          .GLOBL Fill_Rect
22:          .GLOBL Fill_Poly
23:          .GLOBL Mouse_On
24:          .GLOBL Mouse_Off
25:          .GLOBL Mouse_Form
26:          .GLOBL Undraw_Sprite
27:          .GLOBL Draw_Sprite
28:
29:
30:  ; *** LineA-Variablen definieren ***
31:
32:  contrl     equ 4
33:  intin      equ 8
34:  ptsin      equ 12
35:  fgbp0      equ 24
36:  fgbp1      equ 26
37:  fgbp2      equ 28
38:  fgbp4      equ 30
39:  lstlin     equ 32
40:  ln_mask    equ 34
41:  wrt_mode   equ 36
42:  x1         equ 38
43:  y1         equ 40
44:  x2         equ 42
45:  y2         equ 44
46:  pat_ptr    equ 46
47:  pat_msk    equ 50
48:  multifil   equ 52
49:  clip       èqu 54
50:  xmin_clp   equ 56
51:  ymin_clp   equ 58
52:  xmax_clp   equ 60
53:  ymax_clp   equ 62
54:
55:
56:  ; *** Jetzt geht's los! ***
57:
58:
59:
60:  .TEXT
61:
62:
63:  ; ********************************************
64:  ; *                                          *
65:  ; *  Init_LineA initialisiert die LineA-Grafik *
66:  ; *                                          *
67:  ; ********************************************
68:
69:
70:  Init_LineA:
71:    movem.l  d3-d7/a2-a6,-(sp)    ; Register retten
72:    .DC.w $a000                   ; Line-A-Init
                                     aufrufen
73:    move.l   a0,vars              ; Adresse der
                                     LineA-Varliste
74:    movem.l  (sp)+,d3-d7/a2-a6    ; Register
                                     restaurieren
75:    rts                           ; Ab nach Hause
76:
77:
78:    ; ********************************************
79:    ; *                                          *
80:    ; *  Put_Pixel setzt einen Punkt x,y in der  *
81:    ; *  Farbe c                                 *
82:    ; ********************************************
83:
84:
85:  Put_Pixel:
86:    movem.l  d3-d7/a2-a6,-(sp)    ; Register retten
87:    movea.l  vars,a0              ; LineA-Varliste
88:    movea.l  intin(a0),a1         ; Intin-Array
89:    movea.l  ptsin(a0),a2         ; Ptsin-Array
90:    move.w   d0,(a2)              ; x-Koordinate
91:    move.w   d1,2(a2)             ; y-Koordinate
92:    move.w   d2,(a1)              ; Farbe
93:    .DC.w $a001                   ; LineA-Putpixel
94:    movem.l  (sp)+,d3-d7/a2-a6    ; Register
                                     restaurieren
95:    rts                           ; See you later,
                                     alligator !
96:
97:
98:    ; ********************************************
99:    ; *                                          *
100:   ; *  Get_Pixel ermittelt die Farbe des       *
101:   ; *  Punktes x,y                             *
102:   ; ********************************************
103:
104:
105: Get_Pixel:
106:   movem.l  d3-d7/a2-a6,-(sp)    ; Register
                                    retten
107:   movea.l  vars,a0              ; LineA-Varliste
108:   movea.l  ptsin(a0),a1         ; Ptsin-Array
109:   move.w   d0,(a2)              ; x-Koordinate
110:   move.w   d1,2(a2)             ; y-Koordinate
111:   .DC.w $a002                   ; LineA-Getpixel
                                    (Ergebnis in d0)
112:   movem.l  (sp)+,d3-d7/a2-a6    ; Register
                                    restaurieren
113:   rts                           ; Good bye,
                                    crocodile !
114:
115:
116:   ; ********************************************
117:   ; *                                          *
118:   ; *  Line zieht eine Linie vom x1,y1 nach    *
119:   ; *  x2,y2                                   *
120:   ; ********************************************
121:
122: Line:
123:   movea.l  vars,a0              ; LineA-Varliste
124:   move.w   d0,x1(a0)            ; x1
125:   move.w   d1,y1(a0)            ; y1
126:   move.w   d2,x2(a0)            ; x2
127:   move.w   4(sp),y2(a0)         ; y2 vom Stack
128:   move.w   6(sp),fgbp0(a0)      ; Plane 0 vom Stack
129:   move.w   8(sp),fgbp1(a0)      ; Plane 1 vom Stack
130:   move.w   10(sp),fgbp2(a0)     ; Plane 2 vom Stack
131:   move.w   12(sp),fgbp3(a0)     ; Plane 3 vom Stack
132:   move.w   14(sp),ln_mask(a0)   ; LineMask v.Stack
133:   move.w   16(sp),wrt_mode(a0)  ; WriteMode v.Stack
134:   move.w   18(sp),lstlin(a0)    ; LstLine v.Stack
135:   movem.l  d3-d7/a2-a6,-(sp)    ; erst jetzt
                                    Register retten
136:   .DC.w $a003                   ; LineA-DrawLine
137:   movem.l  (sp)+,d3-d7/a2-a6    ; Register
                                    restaurieren
138:   rts                           ; Adios!
139:
140:
141:   ; ********************************************
142:   ; *                                          *
143:   ; *  H_Line zieht eine hor. Linie vom x1,y1  *
144:   ; *  nach x2,y1                              *
145:   ; ********************************************
146:
147:
148: H_Line:
149:   move.l   a0,pp                ; PatternPointer retten
150:   movea.l  vars,a0              ; LineA-Varliste
151:   move.w   d0,x1(a0)            ; x1
152:   move.w   d1,y1(a0)            ; y1
153:   move.w   d2,x2(a0)            ; x2
154:   move.w   4(sp),fgbp0(a0)      ; Plane 0 vom Stack
155:   move.w   6(sp),fgbp1(a0)      ; Plane 1 vom Stack
156:   move.w   8(sp),fgbp2(a0)      ; Plane 2 vom Stack
157:   move.w   10(sp),fgbp3(a0)     ; Plane 3 vom Stack
158:   move.w   12(sp),wrt_mode(a0)  ; WriteMode v.Stack
```

Listing geht weiter...

```
159:    move.l    pp,pat_ptr(a0)       ; PatternPointer
                                          vom Stack
160:    move.w    14(sp),pat_msk(a0) ; PatternMask
                                          vom Stack
161:    movem.l   d3-d7/a2-a6,-(sp)  ; erst jetzt
                                          Register retten
162:    .DC.w $a004                  ; LineA-DrawLine
163:    movem.l    (sp)+,d3-d7/a2-a6  ; Register
                                          restaurieren
164:    rts                          ; Auf Wiedersehen!
165:
166:
167:    ; ********************************************
168:    ; *                                          *
169:    ; *  Filled Rect zeichnet gefülltes Rechteck  *
170:    ; *  (was sonst?)                            *
171:    ; ********************************************
172:
173:
174:  Fill_Rect:
175:    move.l    a0,pp          ; PatternPointer retten
176:    movea.l   vars,a0        ; LineA-Varliste
177:    move.w    d0,x1(a0)      ; x1
178:    move.w    d1,y1(a0)      ; y1
179:    move.w    d2,x2(a0)      ; x2
180:    move.w    4(sp),y2(a0)   ; y2
181:    move.w    6(sp),fgbp0(a0)    ; Plane 0 vom Stack
182:    move.w    8(sp),fgbp1(a0)    ; Plane 1 vom Stack
183:    move.w    10(sp),fgbp2(a0)   ; Plane 2 vom Stack
184:    move.w    12(sp),fgbp3(a0)   ; Plane 3 vom Stack
185:    move.w    14(sp),wrt_mode(a0); WriteMode v.Stack
186:    move.l    pp,pat_ptr(a0)       ; PatternPointer
                                          vom Stack
187:    move.w    16(sp),pat_msk(a0)   ; PatternMask
                                          vom Stack
188:    move.w    18(sp),multifil(a0); Multifill
189:    move.w    20(sp),clip(a0)     ; clip vom Stack
190:    move.w    22(sp),xmin_clp(a0); xminclp vom Stack
191:    move.w    24(sp),ymin_clp(a0); yminclp vom Stack
192:    move.w    26(sp),xmax_clp(a0); xmaxclp vom Stack
193:    move.w    28(sp),ymax_clp(a0); ymaxclp vom Stack
194:    movem.l   d3-d7/a2-a6,-(sp)   ; erst jetzt
                                          Register retten
195:    .DC.w $a005                  ; LineA-FilledRect
196:    movem.l    (sp)+,d3-d7/a2-a6  ; Register
                                          restaurieren
197:    rts                          ; Go Home!
198:
199:
200:
201:    ; ********************************************
202:    ; *                                          *
203:    ; *  Filled Polygon zeichnet ein gefülltes    *
204:    ; *  Polygon                                 *
205:    ; ********************************************
206:
207:
208:  Fill_Poly:
209:    move.l    a0,arrp              ; Koordinaten-
                                          Pointer retten
210:    move.l    a1,pp                ; Pattern-Pointer
                                          retten
211:    movea.l   vars,a0              ; Linea-Varliste
212:    movea.l   contrl(a0),a1        ; contrl[0]
213:    addq.l    #2,a1                ; contrl[1]
214:    move.w    d0,(a1)              ; Anzahl der Punkte
                                          in contrl[1]
215:    addq.w    #1,d0                ; Punkte+1, da
                                          Rückkehr zu
                                          Punkt 0
216:    asl.w     #1,d0                ; mal 2, da 1 Punkt
                                          =2 Koordinaten
217:    movea.l   arrp,a1              ; PointerKoord-
                                          Array
218:    move.l    a2,help              ; A2 retten
219:    movea.l   ptsin(a0),a2         ; Array-Pointer
                                          in ptsin
220:    movea.l   a2,a0                ; A2 retten
221:  loop1:    move.w    (a1)+,(a2)+ ; Kopie User
                                          ->ptsin Array
222:    dbra      d0,loop1             ; Schleifenende
223:    movea.l   help,a2              ; A2 restaurieren
224:    movea.l   vars,a0              ; A0 restaurieren
225:    move.w    d1,y1(a0)            ; y-Wert
226:    move.w    d2,fgbp0(a0)         ; fgbp0
227:    move.w    4(sp),fgbp1(a0)       ; fgbp1
228:    move.w    6(sp),fgbp2(a0)       ; fgbp2
229:    move.w    8(sp),fgbp3(a0)       ; fgbp3
230:    move.w    10(sp),wrt_mode(a0) ; writemode
231:    move.w    12(sp),pat_msk(a0)  ; Pattern Mask
232:    move.w    14(sp),multifil(a0) ; Multifill
233:    move.w    16(sp),clip(a0)     ; clip
234:    move.w    18(sp),xmin_clp(a0) ; xminclp
235:    move.w    20(sp),ymin_clp(a0) ; xmaxclp
236:    move.w    22(sp),xmax_clp(a0) ; yminclp
237:    move.w    24(sp),ymax_clp(a0) ; ymaxclp
238:    move.l    pp,pat_ptr(a0)      ; Pattern Pointer
239:    movem.l   d3-d7/a2-a6,-(sp)   ; Register retten
240:    .DC.w $a006                 ; LineA-Filled Polygon
241:    movem.l    (sp)+,d3-d7/a2-a6   ; Register
                                         restaurieren
242:    rts                           ; Let's go home!
243:
244:
245:    ; ************************************
246:    ; *                                  *
247:    ; *  Mouse_On schaltet die Maus ein  *
248:    ; *                                  *
249:    ; ************************************
250:
251:
252:  Mouse_On:
253:    movem.l   d3-d7/a2-a6,-(sp)   ; Register retten
254:    .DC.w $a009                   ; LineA-Mouse on
255:    movem.l    (sp)+,d3-d7/a2-a6   ; Register
                                         restaurieren
256:    rts                           ; Adios!
257:
258:
259:
260:    ; *************************************
261:    ; *                                   *
262:    ; *  Mouse_Off schaltet die Maus aus  *
263:    ; *                                   *
264:    ; *************************************
265:
266:
267:  Mouse_Off:
268:    movem.l   d3-d7/a2-a6,-(sp)   ; Register retten
269:    .DC.w $a00a                   ; LineA-Mouse off
270:    movem.l    (sp)+,d3-d7/a2-a6   ; Register
                                         restaurieren
271:    rts                           ; Und ab!
272:
273:
274:    ; ************************************
275:    ; *                                  *
276:    ; *  Mouse_Form veraendert die Maus  *
277:    ; *                                  *
278:    ; ************************************
279:
280:
281:  Mouse_Form:
282:    move.l    a0,arrp          ; Pointer Maske sichern
283:    move.l    a1,pp            ; Pointer Cursordaten
                                     sichern
284:    movem.l   d3-d7/a2-a6,-(sp) ;  Register retten
285:    movea.l   vars,a0          ; LineA-Varlist
286:    movea.l   intin(a0),a1     ; intin-Array
287:    move.w    #0,6(a1)         ; intin[3]=0
288:    move.w    #1,8(a1)         ; intin[4]=1
289:    movea.l   arrp,a0          ; Masken-Pointer
290:    adda.l    #10,a1           ; intin[5]
291:    move.w    #15,d0           ; 16 Worte kopieren
292:  loop2:     move.w     (a0)+,(a1)+ ;  Kopie Mask
                                         ->intin[5..20]
293:    dbra      d0,loop2         ; Schleifenende
294:    movea.l   pp,a0            ; Cursordaten-Pointer
295:    move.w    #15,d0           ; 16 Worte kopieren
296:  loop3:     move.w     (a0)+,(a1)+ ; Kopie Cursd
                                         ->intin[21..36]
297:    dbra      d0,loop3         ; Schleifenende
298:    .DC.w $a00b                ; LineA-MouseForm
299:    movem.l   (sp)+,d3-d7/a2-a6 ;  Register
                                        restaurieren
300:    rts                        ; That's it!
301:
302:
303:    ; *************************************
304:    ; *                                   *
305:    ; *  Undraw_Sprite löscht ein Sprite  *
306:    ; *                                   *
```

```
307:    ;  **************************************
308:
309:
310: Undraw_Sprite:
311:    movem.l   d3-d7/a2-a6,-(sp) ; Register retten
312:    movea.l   a0,a2             ; Sicherungsbereich
                                      in A2
313:    .DC.w $a00c               ; LineA-UndrawSprite
314:    movem.l   (sp)+,d3-d7/a2-a6 ; Register
                                      restaurieren
315:    rts                         ; Zurück ins
                                      Hauptprogramm!
316:
317:
318:    ;  **************************************
319:    ;  *                                    *
320:    ;  *  Draw_Sprite zeichnet ein Sprite   *
321:    ;  *                                    *
322:    ;  **************************************
323:
324:
325: Draw_Sprite:
326:    movem.l   d3-d7/a2-a6,-(sp)  ; Register retten
327:    movea.l   a1,a2       ; Sicherungsbereiche in A2
328:    .DC.w $a00d                  ; LineA-DrawSprite
329:    movem.l   (sp)+,d3-d7/a2-a6  ; Register
                                       restaurieren
330:    rts                          ; Good Bye!
331:
332:
333:    ; *** Zwischenspeicher ***
334:
335:            .DATA
336: vars:      .DC.l 1
337: arrp:      .DC.l 1
338: pp:        .DC.l 1
339: help:      .DC.l 1
340:
341:    ;*****     Jetzt ist aber Schluss!       *****
342:            .END
```

Listing 3: Assemblerlisting (GFA-Assembler)

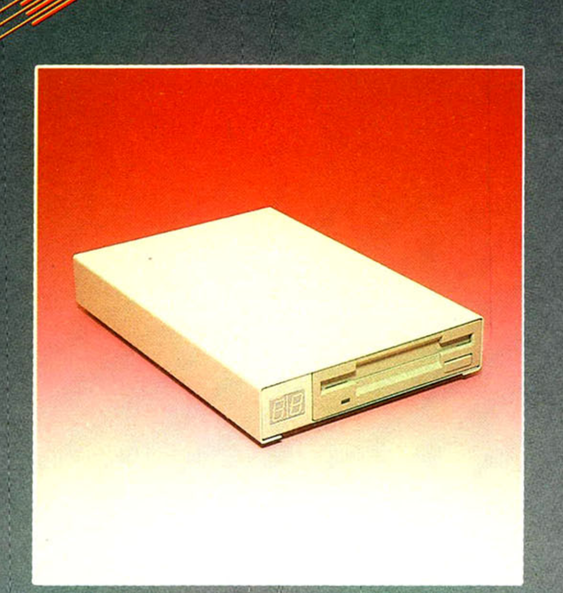
ENDE

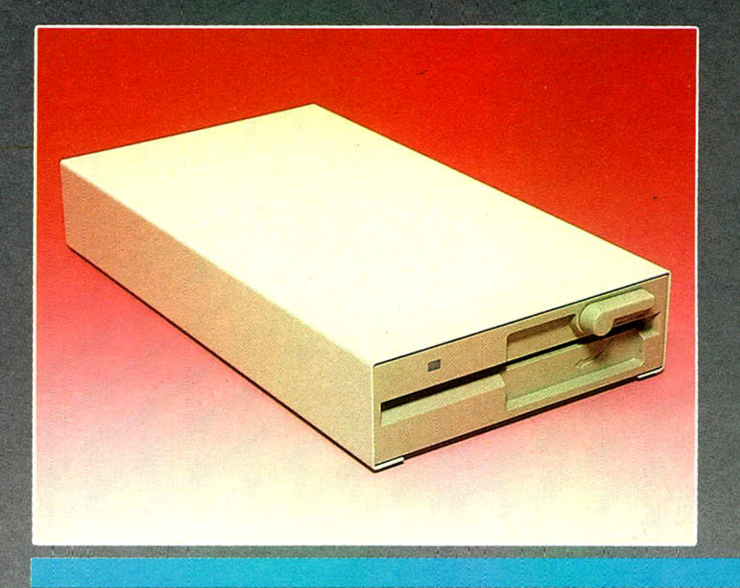


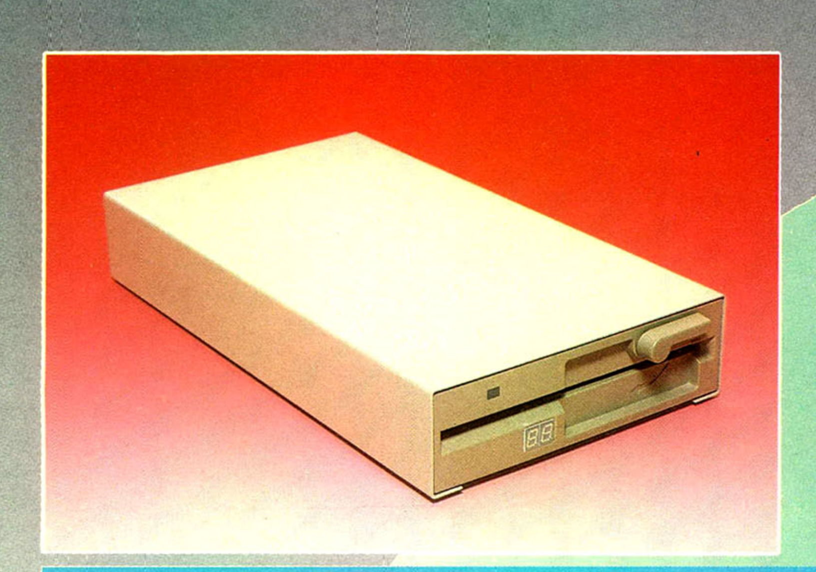


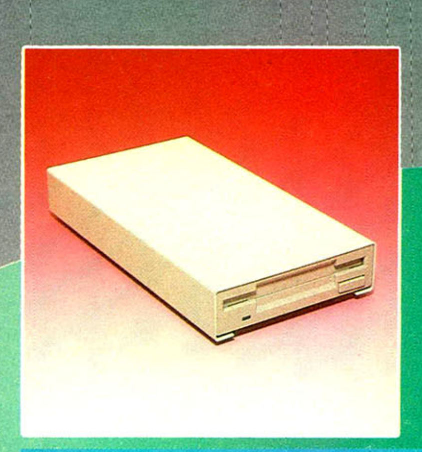
02 31/81 83 25-27
Telefax 02 31/81 74 29
D-4600 Dortmund 1
Burgweg 52 a

GOLEM









Wir liefern im 3-Tage-Rhythmus

Golem für Atari

Qualitätslaufwerke von einem der führenden Hersteller für Computerperipherie
— jetzt auch für alle Atari —

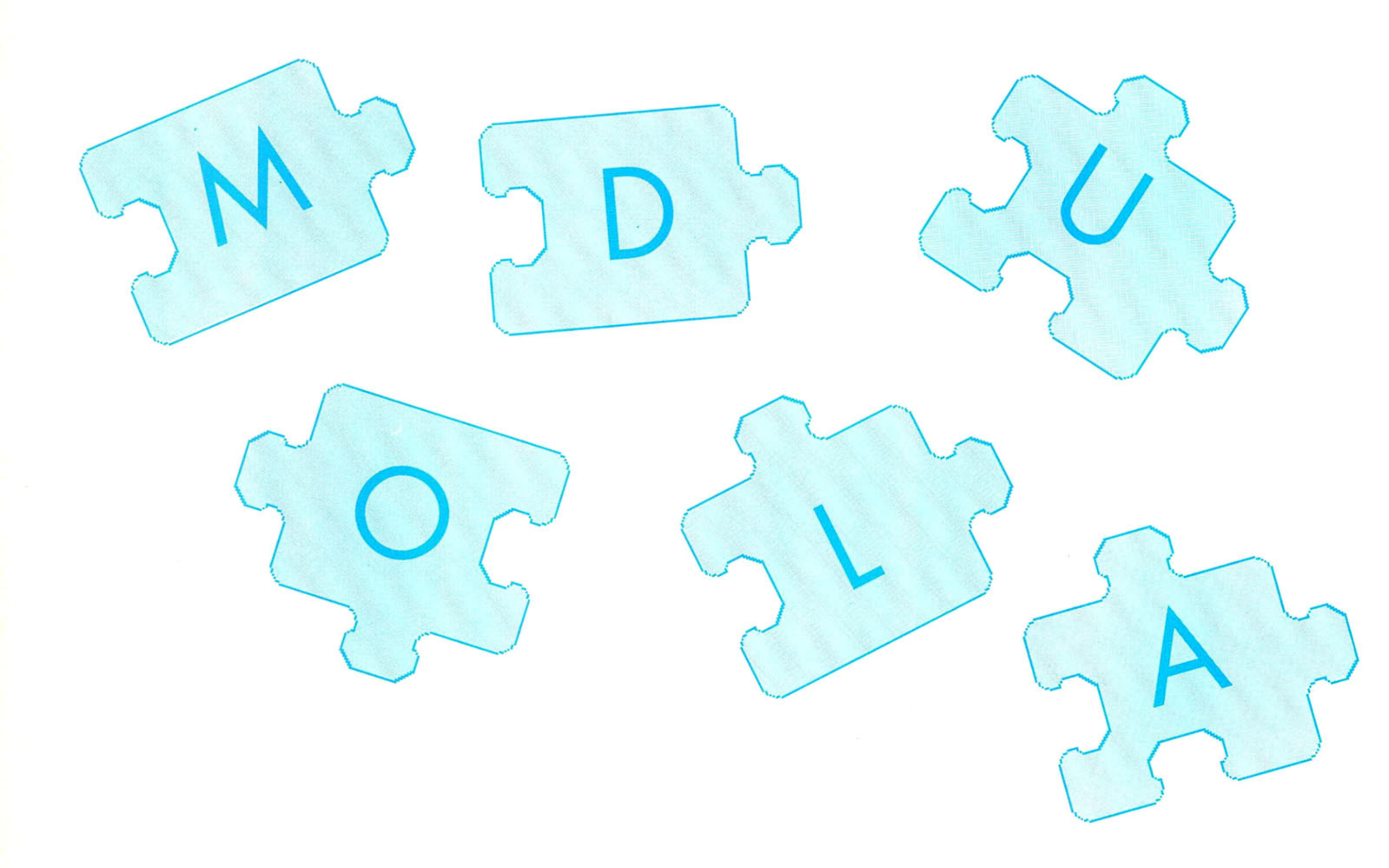
Golem Drive 3,5 Display
NEC 1037a mit heller Frontblende ● Atari-farbenes Metallghäuse ● Abschalter ● Trackdisplay zur aktuellen Spur- und Kopfanzeige **DM 359.-**

Golem Drive 3,5
NEC 1037a mit heller Frontblende ● Atari-farbenes Metallgehäuse ● Abschalter **DM 339.-**

Golem Drive 5,25 Display
NEC Laufwerk mit heller Frontblende ● Atari-farbenes Metallgehäuse ● Abschalter ● 40/80 Track-Umschalter Trackdisplay zur aktuellen Spur- und Kopfanzeige **DM 449.-**

Golem Drive 5,25
NEC Laufwek mit heller Frontblende ● Atari-farbenes Metallgeäuse ● Abschalter ● Abschalter ● 40/80 Track-Umschalter **DM 419.-**

Entwurf + Druck PLUM, Düren-Oberzier, TEL 02428/711



Willkommen beim Modula-2 Kurs in der ST-Computer. Warum ein Kurs und für wen?

Modula-2 hat sich in der letzten Zeit als zweite Entwicklungssprache nach C auf dem ATARI ST etablieren können. Nicht nur die Flut der Compiler, sondern auch deren Qualität machen Modula-2 zu einer Sprache, die für ST-Benutzer immer größere Bedeutung erlangt. In den Ausgaben des vergangenen Jahres konnten Sie Tests der Systeme von TDI, Jefferson, Megamax, Softwave und SPC lesen - und alle zeigten sich als brauchbare Pakete.

Modula-2 ist von den auf Mikros verbreiteten Sprachen die modernste, und das nicht nur aufgrund des Entwicklungsdatums. Konzepte wie Modularisierung und Parallelität bei gleichzeitiger Maschinennähe sind als Merkmale einer Sprache inzwischen unabdingbar. Weder C noch Pascal können hier befriedigen; das Wort Basic braucht man bei der Auswahl einer Hochsprache auch in dieser Hinsicht nicht in den Mund zu nehmen.

Das große Argument der C-Fans, man könne allein mit dieser Sprache das letzte Bit im abgelegensten Speicherwinkel umdrehen, können auch Modula-Programmierer für sich in Anspruch nehmen. Und dabei bleiben Modula-Programme lesbar und sind nicht von Sonderzeichen durchsetzt.

Als Weiterentwicklung von Pascal bietet Modula konzeptuelle Neuheiten, die auch durch das Zauberwort "Turbo" nicht aufgewogen werden können. Wenn man den Vergleich mit Basic überhaupt anstellen will, erkennt man auf den ersten Blick, daß Programme, die über simple Utilities hinausgehen, nur mit einer strukturierten Hochsprache und Modularisierung realisierbar sind - alles Dinge, bei denen BASIC versagt.

Auch wenn Modula aus wissenschaftlichem Blickwinkel schon längst wieder veraltet ist, wird es bei den auf Mikros verbreiteten Sprachen schwer, Argumente dagegen zu finden.

Für wen ist Modula-2 und damit dieser Kurs interessant? Nun, man könnte Hobby-Programmierer einteilen in Anfänger, die grundlegende Programmierkonzepte und Denkweisen kennenlernen wollen, und "alte Hasen", die schon Tausende von Programmzeilen eingetippt haben.

Der Anfänger ist bei Modula-2 aufgrund der Verwandtschaft mit Pascal sehr gut aufgehoben. Diese Sprachen lassen nur einen sauberen Programmierstil zu. Alles muß genau formuliert und bedacht werden. Versuch und Irrtum-Methoden wie bei Basic sind ausgeschlossen, ebenso ist das Tricksen kein Konzept wie bei C.

Der Fortgeschrittene, der lokale Variablen und Rekursion im Traum beherrscht, bekommt mit Modularisierung und Parallelität mächtige Werkzeuge in die Hand, die in keiner der weiter verbreiteten Sprachen zu finden sind. Ihm können sich neue Programmierwelten mit völlig anders gestalteten Algorithmen und einer neuen Entwicklungsweise auftun.

Und das sind die zwei Gruppen, an die sich dieser Kurs wendet. Am Beginn werden für die Einsteiger die grundlegenden Konzepte der Hochsprachen stehen. Stichworte hierzu sind z.B. Deklarationen, Prozeduren, lokale Variablen oder Rekursion. Benutzer anderer Hochsprachen werden zunächst die syntaktischen Unterschiede zu ihrem Lieblingssystem feststellen und dann neue Konzepte kennenlernen, die sicher eine Fülle von Programmideen auslösen.

Was Sie zum Lernen brauchen, ist natürlich ein Modula-System. Fast alle verfügbaren Pakete haben Sie schon in Tests kennengelernt und dabei vielleicht auch

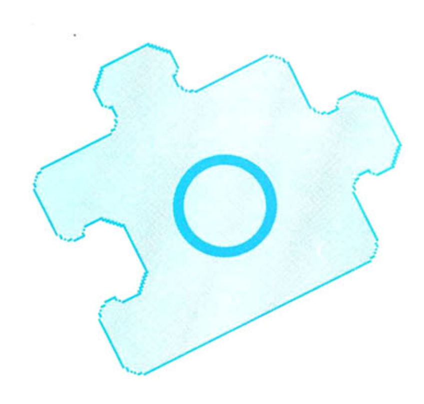
bemerkt, daß es erstaunlich preiswerte Produkte gibt. Sicher wird eines der Angebote Ihren Bedürfnissen entsprechen. Ich werde in diesem Kurs nur Standard-Modula verwenden, so daß alle Beispiele auf allen Compilern lauffähig sind.

Wenn in den einzelnen Artikeln Fragen offen bleiben oder Sie Schwierigkeiten haben, fragen Sie unbedingt nach! Soweit möglich, werden Leserbriefe in einer eigenen Rubrik oder persönlich beantwortet.

Ihre Anregungen sollen vielleicht auch in ein Buch einfließen, das aus dieser Artikelserie entstehen soll. Dabei wird erheblich mehr Platz für Beispiele sein und auch auf Nebenaspekte eingegangen werden.

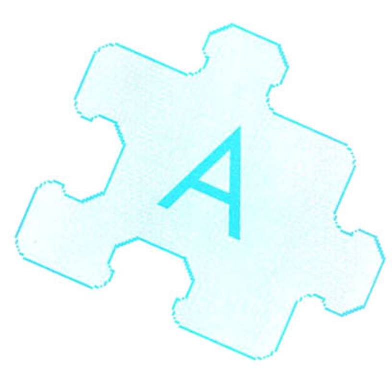
In dieser ersten Folge, die notwendigerweise viel Vorrede enthält, werden Sie noch nicht richtig programmieren. Lernen Sie zunächst die Geschichte von Modula-2 kennen.

Niklaus Wirth, der Entwickler von Pascal, begann im September 1977 an der Eidgenössischen Technischen Hochschule (ETH) in Zürich ein Projekt, in dem eine Workstation und eine dazu passende Sprache entwickelt werden sollte. Diese Sprache war Modula-2 und sollte für die komplette Software des neuen Rechners eingesetzt werden. Die Anwendung reichte dabei vom Betriebssystembau bis hin zu Editoren oder Zeichenprogrammen.



1979 stand dann ein lauffähiger Compiler zur Verfügung: Modula-2 hatte das Licht der Computerwelt erblickt. Der sogenannte ETH-Compiler ist für Entwickler als Sourcecode zur Übertragung auf andere Rechner erhältlich. Auf dem ATARI ST sind TDI-, Jefferson- und SPC-Modula direkte Portierungen dieses Systems.

Der erste Modula-2-Compiler, der auf Mikros auch für Hobby-Anwender erhältlich war, lief unter dem UCSD-System auf dem Apple II. Für den Freak-Computer Gepard war Modula-2 das, was für einen C-64 Basic ist: die mitgelieferte Standardsprache. Das Gepard-Betriebssystem ist ebenfalls in Modula programmiert. Den gleichen Compiler gibt es heute als Megamax-Modula für den ST. Für den Apple Macintosh gibt es eine Portierung des ETH-Systems auch als Public-Domain, leider konnte ich für den ATARI ST nichts Vergleichbares finden.



Die Sprache selber wurde in zwei Revisionen verändert. Dabei war jeweils die Modula-Bibel "Programming in Modula-2" von Niklaus Wirth als Sprachdefinition maßgebend. Das Buch, vom Springer Verlag auch in Deutsch erhältlich (Niklaus Wirth: Programmieren in Modula-2, 3. Auflage, ISBN 3-540-13301-1). Auch wenn es kein Lehrbuch ist und als Buch keine Besonderheiten bietet, ist es jedoch als Referenz so unerläßlich, wie bei C der Kernigham/Ritchie oder für Prolog die Clocksin/Mellish-Definition.

Wenn Sie sich wundern, wo Modula-1 bleibt: Für Experimente mit Parallelprogrammierung entwickelte Wirth die Sprache Modula, über die es jedoch nur universitäre Unterlagen gibt, und die für Hobby-Anwender weder interessant, noch verwendbar ist - eben eine Experimentalsprache. Da die Ergebnisse weiterwendet wurden, übernahm Wirth den Namen für die Universalsprache Modula-2.

Auch wenn es historisch nicht korrekt ist, meint heute jeder Modula-2, wenn er Modula sagt oder schreibt. Und natürlich meine auch ich keine Experimentalsprache, wenn ich mich auf Modula beziehe.

Für den ST war TDI das erste Modula-System, und es blieb lange Zeit das einzige.

Erst in diesem Jahr kamen weitere Systeme auf den Markt. Und wenn Sie eines davon auf Ihrer Diskette oder Festplatte haben, können Sie jetzt das erste Modula-Programm eintippen, an dem Sie die grundsätzliche Struktur der Sprache kennenlernen sollen und einen kleinen Eindruck vom Aussehen eines Modula-Programms bekommen können. Sie finden es als Listing 1.

Das Programm hat keine große Funktionalität, dennoch enthält es schon die hauptsächlichen Elemente eines Modula-Programms. Es handelt sich um das klassische Einführungsprogramm, das die Meldung "Hello world!" auf dem Bildschirm ausgeben soll. Schauen wir uns an, was wir haben.

"MODULE HelloWorld" ist der Programmkopf. Durch ihn wird festgelegt, daß es sich um ein Programm handelt, das den Namen *"HelloWorld"* trägt. Alles, was nun bis zum Gegenstück *"END HelloWorld."* folgt, soll vom Compiler übersetzt werden. In der Modula-Syntax wird dies "Compilation Unit", also "Übersetzungsstück", genannt. Einfacher gesagt, wird durch diese Klammerung der Programmtext begrenzt und durch den Namen benannt.

Als nächstes folgt ein Import. Dahinter steht ein grundlegendes Konzept, die Modularisierung. Für den Anfang denken Sie sich einfach eine Bibliothek, die den Namen *"InOut"* hat, und aus der die vorprogrammierten Prozeduren *"WriteString"*, *"WriteLn"* und *"Read"* geholt werden. Sie werden später noch sehen, daß Module erheblich mehr sind als simple Bibliotheken, für den Anfang soll diese einfache Erklärung reichen. In den Beispielprogrammen werden Sie immer wieder Importe finden, bevor ich zu dem Konzept komme.

Die nächste Einheit in dem Programm ist eine Variablendeklaration. Eine Variable ist etwas, was einen Wert aufnehmen

```
 1: MODULE HelloWorld;
 2:
 3: FROM InOut IMPORT WriteString, WriteLn, Read ;
 4:
 5: VAR ch:CHAR;
 6:
 7: BEGIN
 8: WriteString('Hello world !');
 9: WriteLn;
10: Read(ch);
11: END HelloWorld.
```

Listing 1

kann, und was in Modula einen bestimmten Typ hat. In diesem Fall handelt es sich um den Typ *"CHAR"*, der einem Zeichen entspricht. "ch" ist also ein Kästchen, in das ein Zeichen gelegt werden kann.

Nun beginnt das eigentlich Programm. Alles zwischen *"BEGIN"* und *"END"* bildet einen Block, der ausgeführt werden soll. In dem Block befinden sich hier drei Anweisungen. Eine Anweisung ist etwas, was der Rechner ausführen soll. In diesem Fall sind es Anweisungen, die eine Zeichenkette ausgeben, einen Zeilenvorschub auf den Bildschirm schicken und ein Zeichen einlesen. *"Read(ch);"* ist die Anweisung, ein Zeichen von der Tastatur zu holen und es in die Variable *"ch"* zu packen.

Diese Beschreibung geht natürlich nicht in die Tiefe und läßt zuviel offen, als daß Sie nun ein eigenes Programm schreiben könnten. Denn jedem der Themen, die hier nur in einem Absatz angesprochen sind, sollen und müssen eigene Kapitel gewidmet werden.

Hausaufgaben

Zum Konzept dieser Serie gehören auch Hausaufgaben. Und selbst wenn es sich hier nur um einen Vorspann handelt, sollten Sie in der Zwischenzeit etwas tun. Es gibt drei Aufgaben für Sie:

Die erste Aufgabe ist nicht sehr anstrengend, erfordert allerdings vielleicht etwas Verzicht. Vergessen Sie bitte die BASIC-Programmierung. Vergessen Sie Variablen, die Sie einfach verwenden, ohne sie vorher zu deklarieren. Vergessen Sie die Trial-and-Error-Methode, einfach mal ein Programm zu schreiben und die Fehler vom Interpreter suchen zu lassen. Modula-2 ist eine Hochsprache, die Konzepte benutzt, die in BASIC nicht vorhanden sind, und eine andere Programmierung bewirken.

Wenn Sie sich von diesem Kulturschock erholt haben, fangen Sie bitte mit der zweiten Hausaufgabe an: Lesen Sie die Anleitung zu Ihrem Modula-System. Es nutzt Ihnen wenig, wenn Sie ein Programm zwar eintippen können, dann aber nicht wissen, wie Sie es übersetzen sollen. Handbücher sind da, um gelesen zu werden. Ich kann in diesem Kurs zwangsläufig nicht auf die Besonderheiten bestimmter Systeme eingehen.

Und dann können Sie mit der dritte Aufgabe beginnen, nämlich das kleine Beispielprogramm einzutippen, übersetzen zu lassen und es einmal auszuführen. Wenn das funktioniert hat, versuchen Sie einmal, das Programm so zu ändern, daß es anstelle von "Hello world!" den Text "Hallo Welt!" ausgibt.

In der nächsten Folge werden Sie die Regeln für Bezeichner, also Namen, in Modula kennenlernen und etwas mehr über Variablen, Deklarationen und Typen erfahren. Weiterhin geht es um eine erste Art von Anweisungen, die Zuweisungen, Sie könnten damit Ihr erstes Modula-Programm ohne Vorlage schreiben.

RI

ENDE

***Was bisher geschah*: Dr. Borg hätte es wissen müssen; das Projekt Cee-Dee ließ ihn nicht mehr los. Durch ungezählte Dokumentationen hatte er sich gewühlt, tausendmal das Wörterbuch gewälzt, manche Träne im Kopfkissen zerdrückt. Doch nun wußte er alles über die geheimnisvollen Vorgänge im Kloster Laserheim, er kannte die Rolle des unheimlichen Mister Frame, und sogar die Agenten Reed und Solomon hatte er durchschaut. Alles, was ihm fehlte, war seine Gefährtin Ada Rist, die ihm helfen sollte, der glitzernden Scheibe die Geheimdaten zu entreißen...**



LICHTSPI

Patentes zur Programmierung

Teil III

Ernst beiseite: Ich habe versucht, Ihnen zu zeigen, wie auf einer CD Daten aufgezeichnet werden und wie sie ein CD-ROM davon lesen kann. Sie sollten nun wissen, daß...

- auf einer CD die Bits mittels komplizierter Codierungen auf einer langen Datenspirale abgelegt werden,
- daß die kleinste Dateneinheit ein *Frame* ist, das 24 Datenbytes sowie ein Kontrollbyte (mit acht Subkanalbits) enthält,
- daß aus jeweils 98 Frames ein *Sektor* gebildet wird, der je nach Aufbau 2048 bis 2336 Datenbytes faßt,
- daß pro Sekunde 75 Sektoren (also 75*98=7350 Frames und damit auch 7350 Subkanalbytes) gelesen werden, und daß auf eine CD Daten für mindestens 60 Minuten Spielzeit passen (also mindestens 540 MB).

Der Plan für heute: Die ACSI-Kommandos des CDAR504 werden im Überblick vorgestellt, um Ihnen einen Eindruck von den Fähigkeiten der Kiste zu geben und um Programmierern zum Einstieg zu verhelfen. Schließlich hat das CDAR504 gute Software noch bitter nötig, und je schneller die Programmierer unter Ihnen Bescheid wissen, desto eher dürfen auch die Anwender auf gute Pakete hoffen. "ACSI? Walter?" Auch über den ACSI-Bus erfahren Sie in dieser Folge einige Grundbegriffe, mit denen Sie Ihren Bäcker beeindrucken können. Und *last, but certainly not least* ('tschuldigung, aber mir fällt einfach keine gute deutsche Übersetzung dazu ein) habe ich Ihnen bereits eine komplette Sammlung von Assemblerroutinen zur Einbindung in eigene Programme zu bieten, so daß Sie sich mit den widerlichsten DMA-Interna überhaupt nicht herumschlagen müssen. Ist das nix? Wohlan:

Einsames CD-ROM sucht Anschluß

Das CDAR504 von ATARI hängt am DMA-Port des ST. Auf diesem DMA-Bus hat ATARI ein eigenes Protokoll zur schnellen Datenübertragung definiert: Das ACSI *(ATARI Computer System Interface)*. Es hat einiges vom großen Bruder SCSI (sprich: "skassi"; steht für *Small Computers System Interface*) geerbt, aber leider nicht alles. Wenn Sie die Unterschiede interessieren, schlagen Sie doch mal bei [1] oder [2] nach. Für unsere Zwecke genügt eine Grundausbildung in Sachen ACSI. Anschnallen und das Rauchen einstellen.

Ganz schön fix - Der DMA-Port des ST

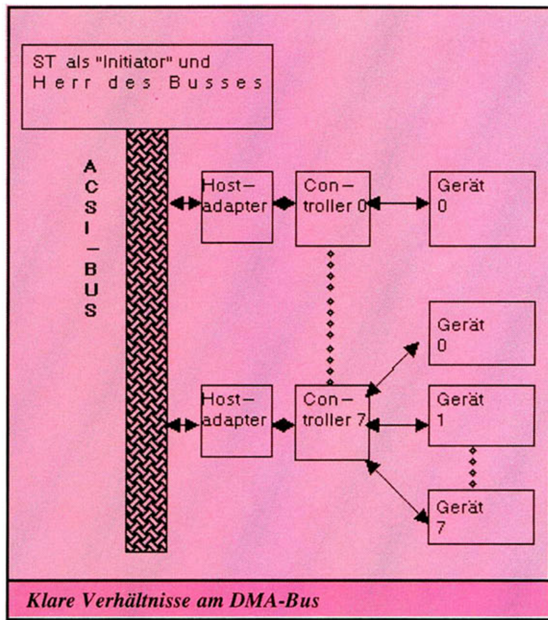
Am DMA-Port des ST, das wissen Sie, hängen Festplatten, Laserdrucker und jetzt auch das CD-ROM. Zur friedlichen Koexistenz braucht man ein Protokoll, an das sich alle halten. Wenn nur ein Quertreiber am Bus Faxen macht, läuft nichts mehr: Das sollte man zum Beispiel bedenken, wenn man sich eine der Billiglösungen für den Festplattenanschluß anschaffen möchte.

Im Prinzip bietet sich am DMA-Bus des ST, der ja eigentlich *ACSI-Bus* heißt, folgendes Bild (Bild 1):

ELE

Nun gibt es ja auch andere, ähnliche Buskonzepte, die ebenfalls den Anschluß vieler Peripheriegeräte erlauben; man denke nur an den IEC-Bus, der ja sogar (rudimentär) beim C64 implementiert war. Wie der ACSI-Bus setzt dieser - allerdings serielle - Bus intelligente Peripheriegeräte voraus, die sich zumindest ihre Schuhe selber binden können und den Rechner nicht mit ihrem eigenen Kram belästigen. Die Unterschiede: Der ACSI-Bus ist ein 8 Bit breiter paralleler Bus, und außerdem wird er von ATARIs DMA-Chip bedient.

Letzteres hat zur Folge, daß auf dem ACSI-Bus Datenübertragungsraten von bis zu 1.3 MByte/Sekunde möglich sind; die Übertragung selbst wird komplett vom DMA-Chip übernommen, der Prozessor bleibt frei.



Klare Verhältnisse am DMA-Bus

Kochrezepte

Sie wissen also nun, was der ACSI-Bus kann, nur noch nicht, wie man's macht. Um eine Übertragung anzuleiern, ist nämlich doch ein wenig Hackerei notwendig, bis man dem DMA-Chip und den beteiligten Controllern seine Wünsche klargemacht hat. Insbesondere ist wichtig, wie die Datenübertragung prinzipiell abläuft:

Der ST herrscht also über den Bus, an dem bis zu 8 verschiedene Zieleinheiten *(Targets)* angeschlossen sein können. Jede dieser Zieleinheiten besitzt einen *Hostadapter*, der die Signale des ACSI-Busses für den *Controller* umsetzt. An diesen Controller können jeweils bis zu 8 *Geräte* angeschlossen sein, je nachdem, wie leistungsfähig der Controller ausgelegt wurde (die Controller in den SH205-Platten von ATARI unterstützen zum Beispiel maximal 2 Plattenlaufwerke). Das heißt: Theoretisch könnte man bis zu 64 Peripheriegeräte per ACSI-Bus an den ST anschließen. Und das ist ja fürs erste schon ganz nett, oder?

1. Der ST wählt ein Zielgerät ("Target") aus, indem ein bestimmtes Signal auf den Bus gegeben wird und danach das erste Byte eines Kommandoblocks, das neben der Kommandonummer auch die Targetnummer enthält.
2. Über den Bus wird der Rest des Kommandos an das Zielgerät geschickt; ein Kommando umfaßt meistens 6, beim CDAR504 auch mal 10 Bytes.
3. Je nach Kommando sendet dann
   - der Rechner Daten an das Zielgerät (z.B. Sektor schreiben),

- das Zielgerät Daten an den Rechner (z.B. Sektor lesen),
- niemand etwas (bei einfachen Kommandos etwa, bei denen die gesamte nötige Information bereitsin den Kommandobytes unterzubringen ist).

4. Nach Beendigung des Kommandos schickt das Zielgerät ein Statusbyte an den Rechner, in dem auch gemeldet wird, ob ein Fehler aufgetreten ist.

Soweit das Prinzip. Wie das im einzelnen vor sich geht, ist eine lange Geschichte und für das folgende nicht dringend notwendig. Es ging mir nur darum, daß Sie mir nicht resigniert abwinken, wenn ich Sie gleich mit Targetnummern (Nummern der Zielgeräte), Kommandoblöcken und Statusbytes bombardiere. Was viel bedeutender ist als die wüste DMA-Hackerei: Was kann man mit dem CDAR504 machen? Welche Kommandos unterstützt es?

Die AUDIO-Kommandos

Damit die CDAR-Besitzer für die nächste Fete gerüstet sind, ziehe ich die Audio-Kommandos vor. Wer die Audio-Anwendung für Spielerei hält, möge bitte diesen Abschnitt überspringen. Wir anderen machen jetzt jedenfalls tüchtig Krach - das CDAR504 ist dafür bestens geeignet:

Audio Start (ACSI-Kommando 6) - Wiedergabe eines Songs starten

Byte	Bitcodierung	
1	T T T 0 0 1 1 0	
2	0 0 0 0 0 0 L R	**L=0: linker Kanal an** **R=0: rechter Kanal an**
3	0 0 0 0 0 0 0 0	
4	S S S S S S S S	**Titelnummer**
5	F F F F F F F F	**Anzahl der Titel**
6	C Q 0 0 0 0 0 0	**"Continue"-/Schnelldurchlauf-Flag (Q)**

Mit diesem Kommando kann man einzelne Titel auf einer Musik-CD anwählen und sie starten. Dabei kann man auch mehrere Titel hintereinander spielen lassen; dazu dient der Parameter im fünften Kommandobyte. *(He! Kim Wilde groovt besser, als ich dachte...)*

Wer's einseitig liebt, kann einzelne Kanäle ausschalten; auch beide natürlich, wenn's Ihnen Spaß macht. Ist das Schnelldurchlauf-Flag gesetzt, werden alle Songs nur jeweils 10 Sekunden angespielt. Im "Continue"-Bit wird angekündigt, daß dieses Kommando fortgesetzt wird und weitere Songnummern folgen. Das heißt also: Man schickt nach dem ersten AUDIO-START-Befehl mindestens einen weiteren nach. Löscht man das "Continue"-Flag im letzten AUDIO-START-Befehl, endet die Programmierung.

Audio Stop ($05) - Wiedergabe beenden

T T T 0 0 1 0 1	**TTT = Targetnummer**
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	

Wahrscheinlich der Lieblingsbefehl meiner Mutter: Er schaltet die Audio-Wiedergabe aus - Funkstille.

Audio Program ($11) - Audio-Wiedergabe programmieren

T T T 1 0 0 0 1	**TTT = Targetnummer**
0 0 0 0 0 0 L R	**L=0: linker Kanal an,** **R=0: rechter Kanal an**
M M M M M M M M	**Startminute in BCD**
S S S S S S S S	**Startsekunde in BCD**
F F F F F F F F	**Startblock in BCD**
M M M M M M M M	**Endminute in BCD**
S S S S S S S S	**Endsekunde in BCD**
F F F F F F F F	**Endblock in BCD**
0 0 0 0 0 0 0 0	
R 0 0 0 0 0 0 0	**Repeat-Flag**

Das ist das raffinierteste Audio-Kommando. Es erlaubt Ihnen, ganz kurze oder auch längere Teile aus einer Musik-CD herauszunehmen und sie wiederzugeben. Die kleinste Einheit ist dabei ein Block, also eine 75tel Sekunde. Das ist im Prinzip genug für allerlei Scratch-Effekte; allerdings sind die Positionierungszeiten beim CD-ROM eben leider zu groß, als daß man wirklich gute und flüssige Tricks damit hinbekommen könnte. Aber vielleicht überraschen uns ja noch pfiffige Programmierer. Das Werkzeug dazu, AUDIO PROGRAM, ist jedenfalls da. Ach ja, bevor ich's vergesse: Setzt man das "Repeat"-Flag, wird der ausgewählte Ausschnitt unendlich oft wiederholt - solange, bis ein AUDIO-STOP-Kommando dem Ringelreih' ein Ende setzt.

Wichtig übrigens bei den Zeitangaben in diesen Kommandos: Wie beim CD-ROM öfters üblich, werden die Zeiten im BCD-Format codiert. BCD steht für *Binary Coded Decimal*; dabei wird ein Byte in *zwei Nibbles* (4 Bits) aufgeteilt, in denen Werte von 0 bis 9 stehen dürfen. Damit sind in einem Byte zwei Dezimalziffern darstellbar (also die Dezimalzahlen von 0 bis 99).

Beispiele:

0 0 0 1 0 1 0 0	= $14	**Codierung für Dezimalzahl 14**
1 0 0 1 0 1 1 1	= $97	**Codierung für Dezimalzahl 97**
0 0 0 0 1 0 1 0	= $0A	**ungültige BCD-Codierung**

Das alles ist ja schon ganz hübsch; aber wie kommt man überhaupt an die Information, welche Titel auf einer CD sind? Triviale Lösung: Auf dem Cover nachsehen. Hackerlösung: Inhaltsverzeichnis der CD einlesen. Inhaltsverzeichnis?

Read TOC ($19) - Inhaltsverzeichnis lesen

T T T 1 1 0 0 1	**TTT = Targetnummer**
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
? 0 0 0 0 0 0 0	**Das Rätsel-über-Rätsel-Flag**

An dieser Stelle ein paar Bemerkungen über die Organisation einer CD: Vor dem eigentlichen Datenbereich, in einem Abschnitt, der etwa 150 Datensektoren entspricht, stehen mysteriöse Daten, an die man mit keinem Lese-Befehl herankommt (schnüff!). Halt - ich nehme fast alles zurück: Dort steht auch das erwähnte Inhaltsverzeichnis, auf neudeutsch auch TOC *(Table of Contents)* - und genau dieses liest der Befehl READ TOC ein. Im TOC wird Buch geführt über die Belegung der CD mit maximal 99 Titeln *(Tracks)*.

Das Inhaltsverzeichnis besteht aus einzelnen Einträgen mit jeweils 4 Bytes:

- Titelnummer
- Minute (0..73)
- Sekunde (0..59)
- Block (0..74)





Unser Service endet nicht an der Ladentür. Auch bei Versandbestellung garantieren wir Ihnen unsere volle Unterstützung.

Aus unserem umfangreichen Software-Angebot:

SPRACHEN	
SPC Modula 2 2 V1.4	398
Lattice C dt. Handb.	298
Aztek C professional	398
Aztek C S. L. Debugger	148
Mark-Williams C V3.0	298
M-W S. L. Debugger	148
Laser C Applications	448
Prospero C neu	248
Torbo C Heimsoeth	198
MCC Pascal 2 dt. Handbuch	248
Pro Pascal V2.1	248
Pro Fortran 77	348
AC Fortran 77	3/8
Salix Prolog 2	198
Omikron Basic Compiler	178
Omikron Assembler	98
GFA Basic Interp. V 3.3	198
Cobol-Paket + PC Ditto	398
OS-9 V 2.2 m div. Compilern	1598
TEXT	
1st Word Plus	198
StarWriter ST Textpr. dt.	198
Steve V3.08 Textpr.	498
WordPerfect Textpr. engl.	798
Signum II Textgestalt.	428
Timeworks DTP V1.11	238
Tempus 2.0 Editor	128
GRAFIK	
Dürer s/w Malprogramm	78
Spektrum 512 farb. Malpr.	148
CADproject V2.0 Normalvers.n	298
CAD-3D Cyberstudio	178
GEM Draw Plus	368
BUSINESS	
Aladin neu V3.0 + Rom's	598
BASICALC Tabellenkalk.	78
K-Graph 3 Grafik u. Stat.	198
Querdruck f. Tabellenkalk.	58
ST-MATH Algebra + Analysis	98
K-Comm 2 Terminalprogramm	148
Logistix integr. Paket	398
1st Adress Dateiverw.	148
dBMAN V 5 Datenbank	598
ST Base III Datenbank	698
T.I.M II Buchhaltung	598
fibuMAN f Buchhaltung	798
BTX-Manager V3.0	428
CRUNCH Harddiskbackup	98
PC-DITTO MS-DOS-Emulator	198
SPIELE	
Psion Chess s/w u. Farbe	69
Elite	69
Dungeon Master F.	69
Carrier Command F.	69
Bolo s/w	69
Gauntlet II F.	69
Pacmania F.	59
Out Run F.	59
Space Quest 2 F.	59
Bubble Bobble F.	59
Guild of Thieves s/w u. F.	49
Chrono Quest F.	89
Flight II Flugsim. s/w u. F.	99
European Scenery Disk	49
Jet Flugsim. F16 und F18	99
HARDWARE	
Einzellaufwerk 3.5''	298
Einzellaufwerk 5.25'' 40/80	398
Speichererweiterung 512K	498
Joystick Commander IV	24
10 Disketten Fuji 3.5'', 2DD	36
10 Disketten 3.5'', 2DD no name	26

Preis- oder Händlerlisten anfordern!
Telefonische Bestellannahme und Hotline-Service: 089/28 12 28
Bei Bestellungen unter DM 200,– beträgt der Versandkostenanteil DM 4,80. Nachnahme DM 3,20. Ins Ausland liefern wir nur gegen Vorkasse (Überweisung oder Euroscheck).

philgerma
Barerstr. 32 · 8000 München 2
☎ 089-281228
Neu: 4600 Dortmund 50
Baroperstr. 337, ☎ 0231-759292

COMPUTER-VERSAND
Schlichting
... der etwas andere Versand

ATARI-Fachmarkt · NEC-Fachhandel · MS-DOS Fachmarkt

24-Stunden-Service

Wir garantieren das jede Bestellung spätestens 24 Stunden nach Eingang unser Haus verläßt, sofern verfügbar

Auf alle gekauften Artikel erhalten Sie natürlich volle Garantie.

Wir führen jede verfügbare Hard- und Software für den Atari ST, sowie alle Bücher

Hier ein kleiner Auszug aus unserem reichhaltigen Programm:

SPIELESOFTWARE:	
Academy	55,–
Afterburner	75,–
Arena	45,–
Balance of Power	85,–
Barbarian II (Psygno.)	70,–
Bolo	60,–
Bolo Werkstatt	65,–
Bubble Ghoast	55,–
Captain Blood	55,–
Leisure suit Larry	60,–
Leviathan	50,–
Lucky Luke	65,–
Mega Pack Compilation	80,–
Metrocross	75,–
Minigolf	55,–
Oids	55,–
Ooze	75,–
Pink Panther	55,–
Psion Chess	65,–
Reisende im Wind II	70,–
Space Quest II	55,–
Star Trek	65,–
Starglider II	65,–
Summer Olympics	60,–
Tetris	50,–
Thrust	40,–
Tolteka	15,–
Triad Compilation	75,–
Ultima IV	80,–
Vectoball	45,–
Veteran	45,–
Virus	45,–
Volleyball Simulator	60,–
Daley Thompson	50,–
Dark Castle	70,–
Dungeon Master	75,–
Elite	65,–
Enduro Racer	40,–
Eye	45,–
Flight Simulator II dt.	85,–
jede Scenery Disc	45,–
Fish	80,–
Football Manager II	60,–
Fred Feuerstein	55,–
Frostbyte	40,–
Gato	75,–
Gauntlet II	70,–
Giana Sisters	55,–
Goldrunner II	45,–
jede Scenery Disc dazu	20,–
Hacker	45,–
Hanse	75,–
Hellowoon	65,–
Hit Disk 1	75,–
Hostages	75,–
Jet	95,–
Jinxter	75,–
Kaiser	120,–
Kampf um die Krone	65,–
Leaderboard Birdie	70,–
1943	55,–

ANWENDERSOFTWARE:	
Aladin Macintosh Enhancer	595,–
Anti Virus Kit	90,–
BS-Fibu	590,–
BS-Handel	490,–
BSS-Plus Module	auf Anfrage
BTX-Manager	400,–
CAD Projekt	ab 290,–
CAD 3D Cyber Studio	175,–
Calamus	390,–
Campus Art	145,–
Campus CAD	790,–
Campus Draft	145,–
Copy Star 3.0	160,–
Daily Mail	175,–
Datamat	90,–
Disk Royal	85,–
Epsimenü	85,–
GEM Desktop 2.2	160,–
GFA-Draft plus	340,–
Systembibliotheken dazu	145,–
Headline Signum Utility	95,–
IPA Degenis III	165,–
Logistix	390,–
ST Pascal plus	240,–
STAD	150,–
Spectrum 512	140,–
Star-Writer	190,–
Star-Writer Lasertreiber	90,–
Steuer Tax '88	90,–
Superbase Professional	590,–
Tempus 2.0	109,–
TIM II Fibu	590,–
Timeworks Publisher	380,–
1st Proportional	90,–
1st Adress	145,–

ZUBEHÖR:	
Staubschutzhauben Kunstleder für:	
SM 124	25,–
ATARI 1040 oder Mega Tast.	je 18,–
ATARI 260/520 ST	15,–
Mega ST Set Monitor + Tastatur	50,–
andere Monitore + Drucker	auf Anfrage
Vortex HD 30	1390,–
Vortex HD 60	1980,–
Handy Scanner	550,–
Monitorumschalter o. Reset	50,–
Mausmatte	18,–
5,25'' ext. Floppy 40/80 Tr.	390,–
3,5'' NO NAME MF 2DD	22,–
3,5'' Maxell MF 2DD	30,–

PUBLIC DOMAIN:
Wir haben über 2.000 Programme auf über 300 Disketten. Nummerierung wie in ST-Computer + eigene.
Außerdem führen wir über 10.000 Programme auf 2.000 Disk. für MS-DOS

JEDE DISKETTE ..... nur 5,– DM
Auch Neuheiten ABO
Fordern Sie unseren Katalog an!

☐ Unseren Gesamtkatalog erhalten Sie kostenlos ☐
Lieferung per NN zzgl. 5,–– DM Versandkosten
Ab 100,–– DM oder Vorauskasse versandkostenfrei
☐ Bestellen Sie bitte schriftlich oder telefonisch ☐

Rund um die Uhr:
☎ 030/7864340

Postanschrift:
Katzbachstraße 8
D-1000 Berlin 61

Ladengeschäft:
Katzbachstraße 6 + 8
D-1000 Berlin 61

FAX: 030/7861904
– Händleranfragen erwünscht –

Titelnummer	Bedeutung
$00	Sie nannten ihn Nobody - so ein Eintrag hat keine Bedeutung.
$A0	Der Titel mit der Nummer, die im Minutenbyte steht, soll als erstes gespielt werden.
$A1	Der Titel mit der Nummer, die im Minutenbyte steht, soll als nächstes gespielt werden.
$A2	Minuten- und Sekundenfeld zeigen die Zeit an, bei der der letzte Titel endet.

Tabelle 1: Spezielle Titelnummern im Inhaltsverzeichnis

Alle Daten sind wieder BCD-codiert. Bei speziellen Titelnummern ändert sich die Bedeutung der restlichen Bytes (Tabelle 1). READ TOC liest einen 512 Byte langen Block mit 128 solchen TOC-Einträgen ein. Setzt man das Flag im letzten Byte, wird die Ordnung der Einträge kräftig durcheinandergeschüttelt; eine bestimmte Strategie konnte ich dabei aber noch nicht ersehen.

Sie sehen: Im Inhaltsverzeichnis steht nichts über die Titel und Interpreten; nur mickrige Startzeiten sind dort verzeichnet. Es fehlt auch eine eindeutige Kennzeichnung der eingelegten CD. Dem könnte man aber mit folgender Strategie abhelfen:

Man lese von allen Musik-CDs (oder auch Daten-CDs, denn die haben auch TOCs) das Inhaltsverzeichnis ein und bilde über die 512 Bytes eine hinreichend sichere Prüfsumme, beispielsweise mit einem CRC-Verfahren [2]. Dazu lasse man sich vom Anwender den Namen der eingelegten CD geben und speichere ihn zusammen mit der Prüfsumme ab. Wenn man nun später eine CD einlegt, liest man zuallererst wieder das Inhaltsverzeichnis, bildet die Prüfsumme und vergleicht sie mit den gespeicherten. Damit kann man dann auch den Titel der CD aus einer Liste heraussuchen. In dieser Liste könnten natürlich auch noch mehr Informationen über die CD stehen, etwa: “Heintje: Mama - vorher Arzt befragen”.

Ein weiteres nützliches Kommando:

Test Unit Ready ($00) - CDAR504 bereit?

```
T T T 0 0 0 0 0     TTT = Targetnummer
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Bit	Bedeutung
0	-
1	ERROR: Fehler aufgetreten
2	BUSY: CDAR504 arbeitet noch ein anderes Kommando ab
3	MEDIA: Platte gewechselt

Tabelle 2: CDAR504 meldet sich zum Rapport.

Wenn gerade noch ein Audio-Kommando abläuft, liefert TEST UNIT READY im Statusbyte ein gesetztes BUSY-Bit. Durch wiederholtes Abfragen kann man so das Ende eines Kommandos relativ genau abpassen. An dieser Stelle sind wohl endlich genauere Informationen über das Statusbyte angebracht. Zum Format siehe Tabelle 2.

Dieses Statusbyte wird natürlich nicht nur nach TEST UNIT READY, sondern nach jedem Kommando geliefert, so daß man vom Erfolg oder Mißerfolg seines Ansinnens rechtzeitig erfährt. Chronisch Neugierige werden das folgende Kommando lieben:

Request Sense ($03) - Statusinformationen holen

```
T T T 0 0 0 1 1     TTT = Targetnummer
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

REQUEST SENSE liefert einen Block von 32 Statusbytes (Tabelle 3).

Byte	Inhalt/Bedeutung
0	$F0
1	$03
2	Fehlernummer: 0 kein Fehler 1 von der Hardware beseitigter Fehler 2 Gerät nicht bereit 3 Prüfsummenfehler 4 Hardware-Fehler 5 ungültiges Kommando 6 keine Platte eingelegt 7 falscher CD-Typ für das betreffende Kommando
3-6:	0
7	$18
8-15:	0

Tabelle 3: Na, das sind ja Zustände - der Statusblock

Diesen Befehl verwendet man vor allem, wenn man im Statusbyte, das man auf ein Kommando hin bekommt, eine allgemeine Fehlerbedingung feststellt. Im Statusblock findet man genauere Informationen zur Fehlerursache. Bei meinem CDAR werden die Bytes 16-31 nicht korrekt gesendet; darum kann ich sie auch noch nicht richtig dokumentieren - ich bitte um Nachsicht. Geplant ist, daß dort Informationen über die aktuelle Position in einem Musikstück und auf der CD gemeldet werden - was ja recht praktisch sein kann, denn mit dieser Information kann man die fehlende Spielzeitanzeige im Display ersetzen.

Die Daten-Kommandos

Read ($08) - logischen Block lesen

```
T T T 0 1 0 0 0     TTT = Targetnummer
0 0 0 H H H H H     Blocknummer, MSB
M M M M M M M M     Blocknummer
L L L L L L L L     Blocknummer, LSB
C C C C C C C C     Blockanzahl (1-255)
0 0 0 0 0 0 0 0
```

Dieses wichtigste Daten-Kommando liest eine Anzahl von *logischen* CD-ROM-Blöcken in den Rechner ein. Die Größe dieser logischen Blöcke stellt man mit MODE SELECT ein - siehe dort. Wichtig: Der erste lesbare Sektor hat die Nummer 150. Warum das so ist, können Sie sich denken, wenn Sie meine entsprechende Bemerkung beim READ-TOC-Befehl rekapitulieren.

Extended Read ($18) - Erweiterte Lesefunktion

T T T 1 1 0 0 0	**TTT = Targetnummer**
0 0 0 0 0 0 0 0	
B B B B B B B B	**Blocknummer (0) oder Minuten in BCD**
B B B B B B B B	**Blocknummer MSB oder Sekunden in BCD**
B B B B B B B B	**Blocknummer**
B B B B B B B B	**Blocknummer LSB oder 0**
0 0 0 0 0 0 0 0	
C C C C C C C C	**Blockanzahl, MSB**
C C C C C C C C	**Blockanzahl, LSB**
0 A 0 0 0 0 0 0	**Audio-Bit**

Bei diesem Lese-Kommando kann die Blocknummer 24 Bit breit sein. Ist das Audio-Bit gesetzt, wird die Blockadresse in Minuten, Sekunden und Blocks aufgeschlüsselt (BCD!).

Mode Select ($15) - Datenmodus setzen

T T T 1 0 1 0 1	**TTT = Targetnummer**
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0	
L L L L P P P P	**Modus (LLLL=logischeBlockgröße, PPPP=physikalische Blockgröße)**
0 0 0 0 0 0 0 0	

MODE SELECT stellt das CDAR504 auf die aktuellen logischen und physikalischen Sektorgrößen ein. Zur physikalischen Sektorgröße (liegt zwischen 2048 und 2336) siehe [3]; das CDAR504 kann aber darüberhinaus einen physikalischen Block in mehrere kleine logische Blöcke unterteilen. Schauen wir uns das Modusbyte und dessen Belegung an. Im oberen Nibble (Einheit von 4 Bits) stellt man die logische, im unteren die physikalische Blockgröße ein (Tabelle 4).

unteres Nibble	**physikalische Sektorgröße**
0	2052/2340 Bytes
1	2048 Bytes
2	2336 Bytes
3	2048 Bytes
4	2336 Bytes
oberes Nibble	**logische Blockgröße**
0	512 Bytes
2	256 Bytes
3	1024 Bytes
4	2048 Bytes
5	2052 Bytes
6	2336 Bytes
7	2340 Bytes

Tabelle 4: Daten in allen Schuhgrößen und Farben der Saison

Die Lese-Kommandos behandeln grundsätzlich nur logische Blöcke! Der Standard-Modus ist übrigens $41 (physikalische und logische Sektorgröße 2048 Bytes).

Paraphernalien

Die weiteren, etwas unwichtigeren Kommandos will ich Ihnen nicht mehr en détail zumuten. Aber kurz erwähnen möchte ich sie noch:

MODE SENSE ($1A):	liefert den aktuellen Modus (per MODE SELECT einstellbar) in einem 16 Byte langen Datenblock
INQUIRE ($12):	Kennung vom CD-ROM holen
MEDIA REMOVAL ($1E):	Plattenauswurf erlauben/ver bieten
SPINDOWN TIMER ($13):	Nachlaufzeit des Drehmotors setzen
SEEK ($0B):	logischen Block anfahren
EXTENDED SEEK ($1B):	logischen Block suchen (die gleiche Erweiterung für SEEK, wie sie EXTENDED READ für READ ist)

Ans Werk

Sie finden das alles ja ganz nett, aber haben partout keine Lust, sich eigene Routinen für die CD-ROM-Ansteuerung zu schreiben? Kann ich verstehen. Aber Sie müssen ja auch nicht selbst zum Assembler greifen. Erstens gibt es von ATARI zum CDAR504 eine XBIOS-Erweiterung, die die wichtigsten Kommandos unterstützt. Und zweitens habe ich mich schon für Sie mit der Programmierung abgekämpft. Das Ergebnis: Eine Sammlung von Assemblerroutinen, die von Hochsprachen aus aufrufbar ist. "Floppyspielereien"-Leser fühlen sich hier zurecht erinnert an LOCKSLEY, den Befreier des Lesekopfes (eine Sammlung von Assemblerroutinen, mit denen man alle Eigenschaften des Floppycontrollers ausnutzen kann). *FIAT_LUX* (Listing 1) ist das CD-Pendant dazu. Alle Informationen über die Ansteuerung dieser Routinen finden Sie im Kasten "Es werde Licht!".

In der nächsten Folge der Lichtspiele - ja, ich bekenne: Ich muß dazu mein ach so durchdachtes Konzept etwas umwerfen - finden Sie dann ein kleines Monitorprogramm in GFA-BASIC, das die FIAT_LUX-Routinen verwendet, um alles aus dem CDAR504 herauszukitzeln.

Vermißtenmeldungen

So eindrucksvoll die Kommandos des CDAR504 auch sind - eines fehlt: Die Möglichkeit, Musik-Daten direkt in den Rechner zu laden, um sie dort weiterzuverarbeiten. Zumindest nach meiner derzeitigen Kenntnis gibt es keinen Weg, an die Edel-Samples auf der Glitzerscheibe heranzukommen; die Lese-Kommandos funktionieren bei Daten-CDs. Oder gibt es einen Trick? Ich glaube nicht, aber das Gerät ist noch neu und unerforscht - wer weiß...

Wenn man dann jedenfalls noch ein Kommando hätte, mit dem man Daten vom Rechner aus direkt auf die Audio-Wandler des CDAR geben könnte, wäre die Kiste wirklich eine Sensation: CD-ROM, CD-Player und 16-Bit-Soundsampler der Spitzenklasse in einem! Hört mich da jemand bei ATARI?

Es scheint auch so, als ob das CDAR504 keine CDs unterstützt, die sowohl Musik als auch Daten beherbergen (das gibt es nämlich!). Auch das ist eine Einschränkung, die ich nicht ganz verstehe. Es gibt jedenfalls CD-ROMs für den PC-Bereich, die solche Fähigkeiten haben.

Sei's drum. Sie wissen jetzt, was der Controller in einem CD-ROM schon alles für Sie leistet: Sektoren lesen, Musikstücke abfahren, Umrechnungen von physikalischen Sektornummern in logische Blöcke und umgekehrt. Auch wenn Sie kein Programmierer sind, können Sie so doch schon immerhin abschätzen, ob

mit dem CD-ROM Anwendungen möglich sind, die für Sie interessant werden könnten.

Hmmm....ich glaube, Sie haben geschummelt. Sie können doch unmöglich das alles gelesen haben - in so kurzer Zeit! Zur Strafe müssen Sie den Artikel jetzt nochmals durchackern:

Was bisher geschah: *Dr. Borg hätte es wissen müssen; das Projekt Cee-Dee ließ ihn nicht mehr los. Durch ungezählte Dokumentationen hatte er sich — [NEIN! Das Heft ist schon voll genug; irgendwann ist Schluß! Die Red.]*

CB

Literatur:

[1] Labude: "Die Festplatte", ST-Computer 1/88-6/88

[2] Brod/Stepper: "SCHEIBENKLEISTER", Eschborn 1988

[3] Brod: "Lichtspiele", Teil II, ST-Computer 11/88

[4] Schmal: CDAR504 Reference Manual, ATARI 1988

Es werde Licht!

FIAT_LUX ist eine Sammlung von Assemblerroutinen, die Sie in Hochsprachen wie C oder BASIC einbinden können; diese Unterroutinen erledigen die etwas kniffligen Arbeiten, die es am DMA-Bus zu verrichten gilt, wenn man seinem CD-ROM mehr als ein paar apathische Fehlermeldungen entlokken will. Die Routinen besitzen einen einheitlichen Einsprungspunkt; will man ein bestimmtes Kommando loswerden, muß man es erst in einer "Kommandostruktur" beschreiben.

Dieser Einsprungspunkt liegt bei der Hauptroutine *"cdinter"*. In BASIC sähe eine Einbindung etwa so aus:

```
Speicher=Gemdos(&H48,L:2000)     ! genügend Speicher
                                 ! per MALLOC
                                 ! besorgen

Bload "fiat_lux.prg",Speicher    ! assembliertes Pro-
                                 ! gramm einladen

Cdinter=Speicher+28              ! Einsprungpunkt ist
                                 ! das erste Byte nach
                                 ! demProgrammheader

Void C:Cdinter(L:Buf%,L:Buf2%,L:Compnt)
                                 ! Aufruf mit drei Para-
                                 ! metern
```

Wohlgemerkt: Für solch eine Einbindung ist eine komplett assemblierte "Programm"-Datei nötig; damit man diese einfach so ohne Relokation einladen kann, werden alle Variablen im Listing *PC-relativ* adressiert (FIAT_LUX ist also voll relokatibel). Der Quelltext ist übrigens auf den MADMAC-Assembler zugeschnitten; für andere Assembler wird man eventuell die Kommentare generell mit Sternchen (*) einleiten müssen.

Für C sollte man dagegen das beim Assemblieren entstehende *Objektfile* mit dem dazugehörigen Anwenderprogramm passend zusammenlinken. Der Einsprungspunkt cdinter ist dazu als global deklariert (je nach Assembler wird man eventuell die *".globl"*-Anweisung in *".xdef"* übersetzen müssen). Der Aufruf im C-Programm:

```
void cdinter(buf,buf2,compnt);
char *buf;
char *buf2;
COMSTRUCT *compnt;
```

Wie Sie dem obigen Aufruf schon ansehen, braucht FIAT_LUX beim Aufruf drei Parameter - und zwar (in dieser Reihenfolge) Zeiger...

... auf einen Sektorpuffer (beim Lesen von Sektoren): *buf*

... auf einen Reservepuffer (beim Lesen von Statusdaten etc.): *buf2*

... auf eine Kommandostruktur: *compnt*

Bleibt noch die Kommandostruktur zu erläutern, auf die *compnt* deutet. In C-Notation sieht sie so aus:

```
typedef struct
    { char comblock[10];   /* Kommandoblock */
      long status;         /* Status-Langwort */
      long timeflag;       /* Timeout-Flag */
      long dmaend;         /* DMA-Endadresse */
    } COMSTRUCT;
```

Für Programmierer, die sich mit C nicht besonders auskennen: Die Kommandostruktur ist ein Speicherblock von 22 Bytes Länge; der Aufbau ist in Tabelle 5 beschrieben.
Nach jedem Kommando setzt FIAT_LUX die Felder 'status', 'timeflag' und 'dmaend' in der Kommandostruktur - wie sich's gehört. Diese Ausgaben werden deswegen in der folgenden Übersicht über die Ein- und Ausgabeparameter bei den einzelnen Kommandes des CDAR504 nicht mehr erwähnt:

REQUEST SENSE

IN : Kommandoblockin der Kommandostruktur Adresse eines mindestens 32 Bytes langen Reservepuffers in *buf2*

OUT : Statusblock (32 Bytes) im Reservepuffer

READ

IN : Kommandoblock in der Kommandostruktur Adresse eines mindestens 4 Bytes langen Reservepuffers in *buf2*; im ersten Langwort des Reservepuffers muß dielogische Blockgröße stehen (also meistens 2048) Adresse eines genügend großen Sektorpuffers in *buf*

OUT : Sektorinhalte im Sektorpuffer

EXTENDED READ

IN : wie bei READ

OUT : ebenfalls wie bei READ

SEEK, EXTENDED SEEK, AUDIO PROGRAM, AUDIO START, AUDIO STOP, TEST UNIT READY, MEDIA REMOVAL, DISK SPINDOWN TIMER

IN : Kommandoblock in der Kommandostruktur

OUT : - (außer Statusbyte etc.)

READ TOC

IN : Kommandoblock in der Kommandostruktur Adresse eines mindestens 512 Bytes langen Puffers in *buf*

OUT : TOC im Puffer

MODE SENSE

IN : Kommandoblock in der KommandostrukturAdresse eines mindestens 16 Bytes langen Puffers in *buf2*

OUT : Modus-Block im Reservepuffer

INQUIRE

IN : wie bei MODE SENSE

OUT : CD-ROM-Kennung im Reservepuffer

Sollten Sie jetzt immer noch auf dem Schlauch stehen, muß ich Sie auf die nächste Folge der Lichtspiele vertrösten. Dort stelle ich einen CD-Monitor in GFA-BASIC vor, der die Anwendung von FIAT_LUX demonstriert. Diesmal hat einfach der Platz im Heft nicht mehr gereicht - FIAT_LUX (Listing 1) ist schon lang genug.

Größe in Bytes	Name	Bedeutung
10	comblock	Hier wird das ACSI-Kommando übergeben, das meistens 6 Bytes, beim CDAR504 aber auch manchmal 10 Bytes lang sein kann (siehe Hauptartikel); die Targetnummer muß vom aufrufenden Programm im ersten Kommandobyte gesetzt werden!
4	status	Ein Langwort, in dem FIAT_LUX nach dem Aufruf einen Statuswert liefert. Im unteren Byte des oberen Wortes steht das Statusbyte, das vom Controller des CD-ROMs nach jedem Kommando versendet wird. Im untersten Byte steht der aktuelle DMA-Status: $00,$02 DMA-Fehler! $01 OK(evt. Daten nicht komplett übertragen) $03 Alles Roger Moore
4	timeflag	Ein Langwort, in dem FIAT_LUX signalisieren kann, daß ein *Timeout* aufgetreten ist; *Timeout* bedeutet: Dasangesprochene Gerät hat nicht innerhalb einer vorgegebenen Zeitspanne auf die Anfrage reagiert - die Routine wurde daher erfolglos abgebrochen. Wenn solch ein Fall auftritt, wird *timeflag* auf -1 ($FFFFFFFF) gesetzt.
4	dmaend	Ein weiteres Langwort, in dem FIAT_LUX am Ende eines Kommandos die aktuelle DMA-Adresse ablegt. Damit kann man beispielsweise kontrollieren, ob eine Übertragung auch komplett stattgefunden hat. Wollte man etwa einen Sektor von 2048 Bytes Länge übertragen und stellt dann fest, daß die Differenz zwischen dem Wert in diesem Langwort und der Startadresse (der Adresse des Puffers also, in dem die betreffenden Daten stehen) 512 ist, dann ist entweder der 1.April, oder wir haben da einen handfesten Fehler.

Tabelle 5: Aufbau der FIAT_LUX-Kommandostruktur

```
 1:  ****************************************
 2:  * FIAT_LUX.S                           *
 3:  * Das Software-Interface zum           *
 4:  * für die "ST-Computer"                *
 5:  * V1.0, 16.11.88, 2 a.m.               *
 6:  * Written & (C) by Claus Brod          *
 7:  *                 Am Felsenkeller 2    *
 8:  *                 8772 Marktheidenfeld *
 9:  *                 09391/3206           *
10:  *                                      *
11:  ****************************************
12:
13:  .globl  cdinter        ; global für Linker
14:
15:  *****************
16:  * Systemvariablen und Konstanten
17:  *****************
18:  dlow    = $ff860d      ; DMA-Adreßzähler,Lowbyte
19:  dmid    = $ff860b      ; DMA-Adreßzähler,Midbyte
20:  dhigh   = $ff8609      ; DMA-Adreßzähler,Highbyte
21:  gpip    = $fffa01      ; GPIP-Register des MFP
22:  daccess = $ff8604      ; DMA-Access-Register
23:  dmodus  = $ff8606      ; DMA-Modus-Register
24:  flock   = $43e         ; Floppy-VBL-Sperre
25:  hz_200  = $4ba         ; 200-Hz-Zähler
26:
```

Listing geht weiter...

```
 27:  STATUS   = 10            ; Offset des Status
                                 in der Kommandostruktur
 28:  TIMEFLAG = 14            ; Offset des Timeout-Flags
 29:  DMAEND = 18              ; Offset der DMA-
                                 Endadresse-Variablen
 30:
 31:  ***************************
 32:  * cdinter: Einsprungspunkt
 33:  * Aktiviert Supervisormodus, analysiert
 34:  * Modus-Langwort und verteilt danach;
 35:  * terminiert nach Umschalten in Usermodus
 36:  * Auf dem Stack wird übergeben:
 37:  * 4(sp)  Zeiger auf Sektorpuffer
 38:  * 8(sp)  Zeiger auf Reservepuffer
 39:  * 12(sp) Zeiger auf Kommandostruktur:
 40:  *         typedef struct {
 41:  *            char comblock[10];
 42:  *            long status;
 43:  *            long timeflag;
 44:  *            long dmaend;
 45:  *            } COMSTRUCT;
 46:  ****************************
 47:  cdinter:
 48:     link a6,#0
 49:     movem.l d0-a5,-(sp) ; Register retten
 50:     move.l 16(a6),a1    ; Zeiger a.Kommandostruktur
 51:     move.l 12(a6),a5    ; Zeiger auf Reservepuffer
                              (buf2)
 52:     move.l 8(a6),a4     ; Zeiger auf Sektorpuffer
                              (buffer)
 53:
 54:     clr.l -(sp)         ; in den Supervisormodus
 55:     move.w #32,-(sp)    ; SUPER
 56:     trap #1             ; GEMDOS
 57:     addq.l #6,sp
 58:     move.l d0,-(sp)     ; alten SP retten
 59:
 60:     clr.l TIMEFLAG(a1)  ; Timeout-Flag löschen
 61:     st flock            ; Floppy-VBL sperren
 62:     moveq.l #0,d0       ; d0.l löschen
 63:     clr.w d3            ; 6-Byte-Kommandos
 64:  * d3.w = 0: 6-Byte-Kommandos
 65:  * d3.w = 1: 10-Byte-Kommandos
 66:
 67:     move.l a1,a0       ;Zeiger auf Kommandoblock
                              kopieren
 68:
 69:  ********************
 70:  * Hier wird der Kommandoblock analysiert
 71:  * IN: D0.L = 0
 72:  *     A0.L ->Kommandoblock
 73:  *     SR   Supervisormodus
 74:  *     Floppy-VBL gesperrt
 75:  ********************
 76:      move.b (a0),d0     ; erstes Kommandobyte holen
 77:      and.w #31,d0       ; nur die unteren 5 Bits
                              interessieren
 78:
 79:      cmp.b #3,d0        ; REQUEST SENSE?
 80:      beq cdreqsense
 81:      cmp.b #8,d0        ; READ?
 82:      beq cdread
 83:      cmp.b #$18,d0      ; EXTENDED READ?
 84:      beq cdextread
 85:      cmp.b #$1b,d0      ; EXTENDED SEEK?
 86:      beq.s cdextseek
 87:      cmp.b #$11,d0      ; AUDIO PROGRAM?
 88:      beq.s cdaudprg
 89:      cmp.b #$19,d0      ; READ TOC?
 90:      beq.s cdrdtoc
 91:      cmp.b #$1a,d0      ; MODE SENSE?
 92:      beq.s cdmdsense
 93:      cmp.b #$12,d0      ; INQUIRE?
 94:      beq.s cdinquire
 95:
 96:  * alles andere ist stdcommand
 97:
 98:  ************************
 99:  * stdcommand: Schickt Standard-Kommandoblock
100:  * IN: a0.l Zeiger auf comblock
101:  ************************
102:  stdcommand:
103:     move.w #$8a,d1      ; DMA-Modus
104:     bsr sd1toend        ; Kommandoblock schicken
105:
106:     move.w #$8a,dmodus ; ACSI-Bus selektieren
107:     bsr getstatus          ; Status holen
108:
109:  ***********************
110:  * exitus: Interface beendet die Arbeit
111:  ***********************
112:  exitus:
113:     clr flock              ; Floppy-VBL freigeben
114:
115:     move.w #32,-(sp)       ; SUPER
116:     trap #1                ; GEMDOS
117:     addq.l #6,sp
118:     movem.l (sp)+,d0-a5    ; Register holen
119:     unlk a6
120:     rts                    ; raus hier
121:
122:  *********************
123:  * cdextseek: EXTENDED SEEK für CD
124:  * cdaudprg: AUDIO PROGRAM für CD
125:  *********************
126:  cdextseek:
127:  cdaudprg:
128:     moveq.l #1,d3           ; 10-Byte-Kommando
129:     bra stdcommand
130:
131:  *********************
132:  * cdrdtoc: READ TOC für CD
133:  *********************
134:  cdrdtoc:
135:     moveq.l #1,d5          ; 1 Block übertragen
136:     move.l a4,d6           ; ->Puffer
137:     bra rdentry            ; und lesen
138:
139:  *********************
140:  * cdmdsense: MODE SENSE für CD
141:  * cdinquire: INQUIRE
142:  *********************
143:  cdmdsense:
144:  cdinquire:
145:     moveq.l #1,d5          ; 1 Block übertragen
146:     move.l a5,d6           ; ->Reservepuffer
147:     bra rdentry            ; und los geht's
148:
149:  *********************
150:  * calcblk: Errechnet die Anzahl der
151:  * zu übertragenden DMA-Blocks (512 Bytes)
152:  * IN: d0.w Blockanzahl (aus Kommando)
153:  *     a5 Zeiger auf logische BlockgröPe in Bytes
154:  *     (Langwort, steht am Anfang des
                 Reservepuffers)
155:  * OUT: d5.w Anzahl der DMA-Blocks
156:  * USES: d6.l
157:  *********************
158:  calcblk:
159:     move.l (a5),d5        ; Blockgröße holen (steht
                                 in buf2)
160:     mulu.w d0,d5          ; mal Blockanzahl
161:
162:     divu   #512,d5        ; in DMA-Blocks umrechnen
163:     move.l d5,d6          ; Ergebnis nach d6
164:     swap   d6             ; Worthälften tauschen
165:     tst.w  d6             ; Division mit Rest?
166:     beq.s  calc1          ; nein
167:     addq.w #1,d5          ; ein Block mehr
168:  calc1:
169:     rts
170:
171:  *********************
172:  * cdreqsense: REQUEST SENSE für CD
173:  * IN: A0.L -> Kommandoblock
174:  *********************
175:  cdreqsense:
176:     bsr rdtoggle          ;DMA-Status/Puffer löschen
177:     move.w #1,daccess     ;max. 1 DMA-Block
                                übertragen
178:
179:     move.l a5,d0           ; Startadresse des
                                  Statuspuffers
180:     bsr setdma             ; in DMA-Zählerregister
181:
182:  rqlp:
183:     move.w #$8a,d1    ; Moduswort $8A
184:     clr.w d0          ; d0.w löschen
185:     bsr sd1tox        ; die ersten x Kommandobytes
186:     bmi rwerr         ; Timeout?
187:
```

Listing geht weiter...

```
188:     move.l #$0a,daccess  ; letztes Kommandobyte,
                                DMA starten
189:     bsr waitforcom       ; Übertragung abwarten
190:     bmi rwerr            ; Timeout?
191:
192:     move.w #$8a,dmodus   ; Bus selektieren
193:     nop
194:     move.w daccess,d0    ; CS low, ACK bleibt
                                High,R/W auf R
195:
196:     moveq.l #130,d2      ; 100 reicht auch
197: as_time_goes_by:
198:     dbf d2,as_time_goes_by  ; Zeitschleife
199:
200:     move.w #$8a,dmodus   ; Bus selektieren
201:
202:     bsr getst2           ; Status holen
203:     bra exitus
204:
205: **********************
206: * cdextread: EXTENDED READ für CD
207: **********************
208: cdextread:
209:     moveq.l #0,d0
210:     move.b 7(a0),d0      ; MSB der Blockanzahl
211:     lsl.w  #8,d0         ; mal 256
212:     move.b 8(a0),d0      ; LSB der Blockanzahl
213:     moveq.l #1,d3        ; 10-Byte-Kommando
214:
215:     bra.s cdr1           ; weiter wie bei READ
216:
217: **********************
218: * cdread: READ für CD
219: * IN: A0.L -> Kommandoblock
220: **********************
221: cdread:
222:     moveq.l #0,d0
223:     move.b 4(a0),d0      ; Blockanzahl holen
224: cdr1:
225:     bsr calcblk          ; Anzahl der DMA-Blocks
                                ausrechnen
226:     move.l a4,d6         ; -> Puffer
227:
228: rdentry:
229:     move.w #$88,dmodus   ; Busanforderung
230:     moveq.l #0,d0        ; d0 löschen
231:     move.b (a0),d0       ; Kommando und Target
                                holen
232:     swap d0              ; Worthälften vertauschen
233:     move.w #$8a,d0       ; Modus einkopieren
234:     move.l d0,daccess    ; und alles auf den Bus
235:     bsr zeiteisen        ; auf Bestätigung warten
236:     bmi.s rwerr          ; Timeout?
237:
238:     move.l d6,d0         ; Pufferadresse
239:     bsr setdma           ; als DMA-Startadresse
240:     move.w #$8a,d1       ; DMA-Modus
241:     bsr.s sd2tox         ; Kommandobytes 2 bis x
                                übergeben
242:     bmi.s rwerr          ; Fehler?
243:
244:     bsr rdtoggle
245: rdcnt:
246:     move.w d5,daccess    ; Sektoranzahl an
                                DMA-Sektorzähler
247:     nop
248:     move.w #$8a,dmodus   ; HDC wieder selektieren
249:     nop
250:     move.l #$a,daccess   ; Übertragung starten
251:     bsr waitforcom       ; auf Kommandoende warten
252:
253:     move.w #$8a,dmodus   ; HDC selektieren
254:     nop
255:
256: rwerr:
257:     bsr getstatus        ; Status holen
258:     bra exitus           ; und raus
259:
260: **********************
261: * sd2tox: Schickt die Kommandobytes 2 bis x
262: * auf den ACSI-Bus (per sdbyte)
263: * IN: a0.l Zeiger auf comblock
264: *     d1.w DMA-Modus
265: *     d3.w = 0: 6-Byte-Kommando (x=5)
266: *          = 1: 10-Byte-Kommando (x=9)
267: * OUT: N-Flag gesetzt, wenn Timeout
268: **********************
269: sd2tox:
270:     movem.l d0/d2/d4,-(sp)    ; d0,d2,d4 retten
271:     moveq.l #0,d0             ; d0.l löschen
272:     clr.w d2                  ; d2.w initialisieren
273:     moveq.l #4,d4
274:     tst.w d3                  ; 10-Byte-Kommando?
275:     beq.s sdloop
276:     moveq.l #8,d4
277:
278: sdloop:
279:     addq.w #1,d2          ; d2++
280:     move.b 0(a0,d2),d0    ; Kommandobyte holen
281:     bsr.s sdbyte          ; und auf Bus legen
282:     bmi.s sd2err          ; Timeout?
283:
284:     cmp.w d4,d2           ; schon x-1 Bytes
                                übertragen?
285:     bne sdloop            ; nein, weitermachen
286:
287: sd2err:
288:     movem.l (sp)+,d0/d2/d4  ; Register zurückholen
289:     rts                     ; und raus
290:
291: **********************
292: * sd1tox: Schickt Kommandobytes 1 bis x a.ACSI-Bus
293: * (per sdbyte)
294: * IN: a0.l Zeiger auf comblock
295: *     d1.w DMA-Modus
296: *     d3 = 0: 6-Byte-Kommando
297: *     d3 = 1: 10-Byte-Kommando
298: * OUT: N-Flag gesetzt, wenn Timeout
299: **********************
300: sd1tox:
301:     clr.w d0              ; d0.w löschen
302:     move.w #$88,dmodus    ; ACSI-Bus wachrütteln
303:     move.b (a0),d0        ; erstes Kommandobyte holen
304:     bsr.s sdbyte          ; und auf ACSI-Bus legen
305:     bmi.s sd1err          ; Timeout?
306:     bsr sd2tox            ; Kommandobytes 2 bis x
                                schicken
307: sd1err:
308:     rts
309:
310: **********************
311: * sd1toend: Schickt Kommandobytes 1 bis 6 bzw. 10
             auf ACSI-Bus
312: * (per sdbyte)
313: * IN: a0.l Zeiger auf comblock
314: *     d1.w DMA-Modus
315: *     d3 = 0: 6-Byte-Kommando
316: *     d3 = 1: 10-Byte-Kommando
317: * OUT: N-Flag für Timeout
318: **********************
319: sd1toend:
320:     bsr sd1tox            ; Kommandobytes 1 bis x
                                schicken
321:     bmi.s sderr           ; Timeout?
322:     move.b 5(a0),d0       ; letztes Kommandobyte holen
323:     tst.w d3              ; 10-Byte-Kommando?
324:     beq.s sdlast          ; nein, weiter
325:     move.b 9(a0),d0       ; letztes Kommandobyte holen
326: sdlast:
327:     swap d0               ; Worthälften vertauschen
328:     move.w d1,d0          ; DMA-Modus
329:     move.l d0,daccess     ; auf Bus damit
330:     bsr.s waitforcom      ; auf Kommandoende warten
331: sderr:
332:     rts
333:
334: **********************
335: * sdbyte: Byte in d0 an HDC schicken
336: * IN:  d0.l wie folgt:
337: *       ___________
338: *       | X.b | X.b | 0.b | CB.b|
339: *       ___________
340: *       CB=Commandbyte
341: *       d1.w DMA-Modus ($8A/$18A)
342: **********************
343: sdbyte:
344:     swap d0               ; Worthälften vertauschen
345:     move.w d1,d0          ; DMA-Modus einschieben
346:     move.l d0,daccess     ; Byte auf Bus und Modus
                                an DMA-Chip
347:                           ; drop-thru...
348: **********************
```

```
349:  * zeiteisen: Wartet auf IRQ des HDC
350:  * USES: d0,d1
351:  *********************
352:  zeiteisen:
353:     movem.l d0/d1,-(sp)
354:
355:     moveq.l #0,d0          ; Fehler-Default
356:     moveq.l #20,d1         ; 20 Ticks warten
357:  gettimer:
358:     add.l hz_200,d1        ; d1 plus 200Hz-Zähler
359:  zeita:
360:     btst #5,gpip           ; auf HDC-IRQ testen
361:     beq.s fix_und_fertig ; ist angekommen, fertig
362:     cmp.l hz_200,d1        ; Timer-Zielwert erreicht?
363:     bne zeita              ; nein, weiter
364:
365:     moveq #-1,d0            ; Fehler übermitteln
366:     move.l d0,TIMEFLAG(a1)  ; Timeout-Flag setzen
367:  fix_und_fertig:
368:     tst.l d0                ; N-Flag aktualisieren
369:     movem.l (sp)+,d0/d1
370:     rts                     ; und raus
371:
372:  **********************
373:  * waitforcom: Wartet nach Kommandoblock auf
              Kommandoende
374:  * USES: d0,d1,a2
375:  **********************
376:  waitforcom:
377:     movem.l d0/d1,-(sp)
378:     moveq.l #0,d0                ; Default-Fehler
379:     move.l #800,d1               ; langes Timeout
380:     bra.s gettimer               ; und warten
381:
382:  ********************
383:  * setdma: DMA-Startadresse setzen
384:  * IN: D0.L Pufferadresse
385:  ********************
386:  setdma:
387:     move.b d0,dlow               ; Lowbyte
388:     asr.l #8,d0
389:     move.b d0,dmid               ; Midbyte
390:     asr.l #8,d0
391:     move.b d0,dhigh              ; Highbyte
392:     rts
393:
394:  ********************
395:  * getdma: DMA-Adresse lesen und in dmaend
396:  * ablegen
397:  ********************
398:  getdma:
399:     clr.w d0
400:     move.b dhigh,d0         ; Highbyte lesen
401:     asl.l #8,d0             ; mal 256
402:     move.b dmid,d0          ; Midbyte
403:     asl.l #8,d0             ; mal 256
404:     move.b dlow,d0          ; Lowbyte
405:     move.l d0,DMAEND(a1) ; DMA-Adresse ablegen
406:     rts                     ; und raus
407:
408:  *********************
409:  * rdtoggle: DMA-Puffer und DMA-Status löschen
410:  * Sektorzähler selektieren, Datenrichtung auf
     lesen
411:  *********************
412:  rdtoggle:
413:     move.w #$98,dmodus
414:     nop
415:     move.w #$198,dmodus
416:     nop
417:     move.w #$98,dmodus
418:     rts
419:
420:  **************************
421:  * getstatus: Liest Statusbyte
422:  * deselektiert HDC
423:  * ACSI-Bus muP bereits selektiert sein ($8a oder
     $18a nach dmodus)
424:  * OUT: Status in status
425:  * USES: d0,a2
426:  **************************
427:  getstatus:
428:     move.w daccess,d0       ; Status holen
429:
430:  getst2:
431:     swap d0                 ; Worthälften tauschen
432:     move.w dmodus,d0        ; DMA-Status holen
433:     and.l #$ff0007,d0       ; ausmaskieren
434:     move.l d0,STATUS(a1) ; in status ablegen
435:
436:     move.w #$80,dmodus      ; FDC selektieren
437:     bra getdma              ; DMA-Adresse holen
```

Listing: FIAT_LUX

ENDE

Nicht schon wieder Malprogramme

Mal- und Zeichenprogramme auf dem ATARI ST gibt es fast schon wie Sand am Meer - und doch kommen immer wieder neue auf den Markt!

Was war der Freak zu Beginn des ATARI ST-Zeitalters doch froh, wenn er wenigstens ein paar Linien mühsam mit dem mitgelieferten ST-BASIC auf den Bildschirm brachte. Zu jener Zeit war ein komfortables Malprogramm noch Mangelware. Dies erkannten die Softwarehäuser und reagierten natürlich binnen kürzester Zeit darauf. Als die ersten fleißigen Grafikhelfer erschienen, hatten sie einen entscheidenden Nachteil: anfangs konnte man sie wegen der Pseudo-Monopolstellung der Hersteller kaum bezahlen. Am glücklichsten war damals sicher der Besitzer des Farbmonitors SM1224. Schließlich konnte er das Public Domain-Programm *"NEOCHROME"* zum Nulltarif beziehen und so seinen Geldbeutel schonen. Ebenso spärlich wie der Preis waren auch die Funktionen: Ein paar Linien und Rechtecke, vielleicht gerade noch ein Kreis. An das Niveau, das man von heutiger Software gewohnt ist (Ellipsenausschnitte, beliebige Zeichensätze, Lasso etc.), war keinesfalls zu denken.

Nach und nach erschienen immer mehr Grafikpakete auf dem Markt, und so fielen auch die Preise auf einen erträglichen Level. Inzwischen sind gute Zeichenprogramme für unter 100,- DM erhältlich, die (fast) alle wünschenswerten Funktionen in sich vereinen.

Ständig strömt auf dem Grafiksektor neue Software auf den Markt, wer soll sich denn da noch auskennen??? So geht es sicher den meisten Computerbesitzern, denen der weite Software-Dschungel undurchdringlich erscheint. Schließlich läßt es normalerweise der Geldbeutel kaum zu, alle Programme zu kaufen, schon deshalb, weil eine Rückgabe meist ausgeschlossen ist. Und welcher Händler ist so großzügig, mehrere Programme über's Wochenende auszuleihen, damit der Kunde das Programm ermitteln kann, das für seine Zwecke am besten geeignet ist?

Um einen kleinen Überblick über die interessanteste Zeichen-Software der letzten Zeit zu geben, die sich größtenteils in einem lohnenswerten Preis-Leistungsverhältnis befindet, haben wir einige von diesen Programme getestet:

STARPAINTER ST,
LAVADRAW III (3.0),
GAMMA_03,
ARTKRAFT (V1.31),
PUBLIC PAINTER (V0.45)
und SYMPATIC PAINT (V1.40).

Ursprünglich sollte noch MEGAPAINT von Tommy Software mit in die Auswahl genommen werden, aber die neue Version 2 lag uns bei Redaktionsschluß leider noch nicht vor.

In erster Linie waren uns zwei Eigenschaften wichtig: die Bedienfreundlichkeit und die Funktionen, die das Programm bietet. Bedeutend erschien uns, ob es sich bedienen ließ, ohne daß stundenlang das Handbuch gewälzt werden mußte. Bewertet wurde auch, ob die Menüs gut miteinander kombiniert waren, oder ob ein Zeichenvorgang eher in ein wildes "Hin- und Her-Gefummle" mit der Maus ausartete. Wir prüften, ob es nur die Standardfunktionen gibt und was die zusätzlich eingebauten taugen. Es war zu beurteilen, ob sie für den "Otto-Normal-Verbraucher" eher unnütz oder wirklich hilfreich waren. Gegenstand des Tests war auch, ob das Programm mit einer Festplatte zusammenarbeitet, wieviel Bildschirme dem Zeichner zur Verfügung stehen, welche Grafikformate eingelesen werden können und vieles andere mehr.

STARPAINTER ST

"Das professionelle Grafikprogramm + Handbuch", so wirbt der Hersteller SY-

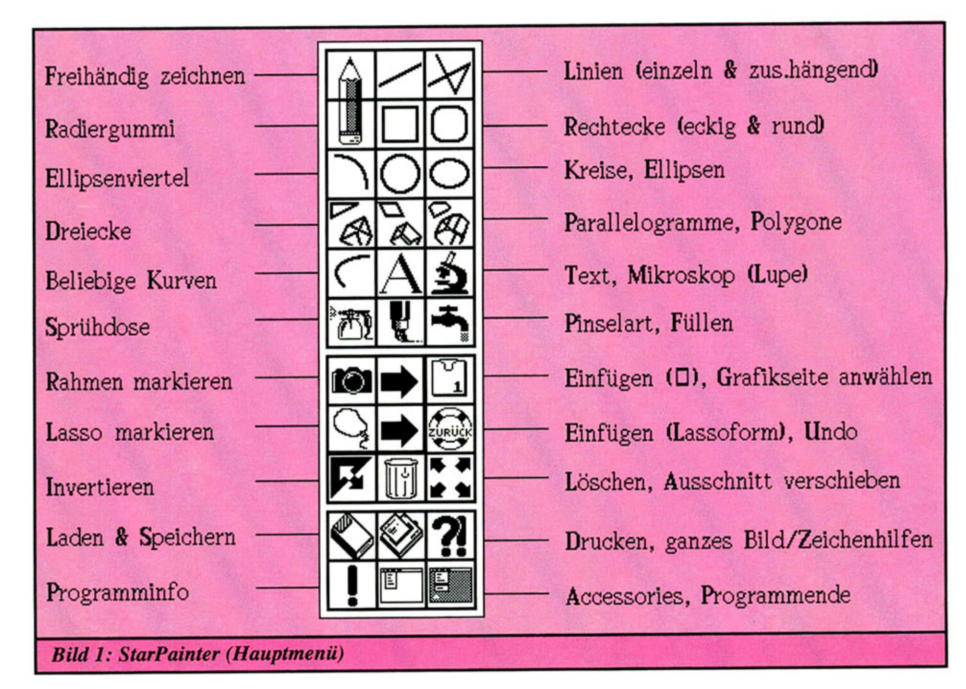


Bild 1: StarPainter (Hauptmenü)

BEX auf der Verpackung des StarPainters. Öffnet der Käufer die Verpackung, die der einer Videokassette ähnlich ist, fällt zunächst einmal das professionell gebundene und gedruckte Handbuch ins Auge, das mit 134 Seiten das zweitumfangreichste unter den "Kandidaten" ist. Mag nun jemand meinen: "Die Qualität ist entscheidend und nicht die Quantität", so läßt sich nur sagen, daß das, was in diesem Handbuch dem Benutzer geboten wird, nicht nur in puncto Umfang die Mitkonkurrenten übertrifft, sondern auch in bezug auf den Inhalt. In einer knappen, 32-seitigen Übersicht werden die wichtigsten Funktionen nach ausführlicher Erläuterung im Hauptteil noch einmal kurz zusammengefaßt, und damit kann die Übersicht wunderbar als Nachschlagewerk benutzt werden.

Das Programm scheint auf den ersten Eindruck etwas knapp an Funktionen zu sein, die über den Standard hinausgehen, doch bei genauerem Betrachten des Handbuches merkt der Leser schnell, daß er sich schwer getäuscht hat. Es ist auf jeden Fall empfehlenswert, das Handbuch durchzuarbeiten, selbst wenn sich das Programm auch "ohne" leicht bedienen läßt, denn sonst geht doch so mancher Trick am StarPainter-Benutzer vorbei.

Das Programm bietet eigentlich alle Funktionen, die zum "täglichen Zeichnen" nötig sind: neben Standardfunktionen wie Linie, Kreis, Freihand, Rechteck, Ellipse und GEM-Text gibt es die Sprühdose, Viertelellipsen, ladbare Zeichensätze, ein Lasso zum Ausschneiden und weitere Funktionen, ja sogar dreidimensionale Parallelogramme, Dreiecke und Polygone. Auch Verzerren, Bilden einer Sinuskurve aus einem Ausschnitt, Spiegeln und Invertieren wurden nicht vergessen. Außerdem existiert eine gute Lupe: es lassen sich drei Vergrößerungsstufen wählen, und ohne Verlassen der Lupe kann der ganze Bildschirm abgetastet werden. Die Lupe ist schnell und bietet verschiedene Funktionen: Punkte setzen/löschen, Füllen, Invertieren und UNDO. Das Hauptmenü von StarPainter ST ist in Bild 1 zu sehen.

Die Besitzer des Grafiktabletts der Firma CRP-Koruk können sich besonders freuen, denn die Entwickler bei Sybex haben hierfür eine spezielle Programmanpassung integriert. Gedruckt werden kann sowohl auf 9-Nadeldruckern als auch auf 24-Nadlern, und zwar entweder ein Bildausschnitt oder der ganze Doppelbildschirm (eine DIN A4-Seite). Sollten Probleme mit der Anpassung auftauchen, so lassen sich auch hier die Parameter einstellen. Bei 24 Nadeln stehen zwei Druckgrößen zur Verfügung, bei 9 Nadeln kann nur zwischen Schnell- und Schöndruck gewählt werden. Bildformate anderer Programme werden beim Laden automatisch erkannt und umfassen bis auf "STAD-gepackt" alle gängigen. Außerdem kann im Falle eines Falles mit Tricks versucht werden, unbekannte Bilder, die nicht dem Standard entsprechen, mit Konvertierung usw. doch noch auf den Schirm zu bringen.

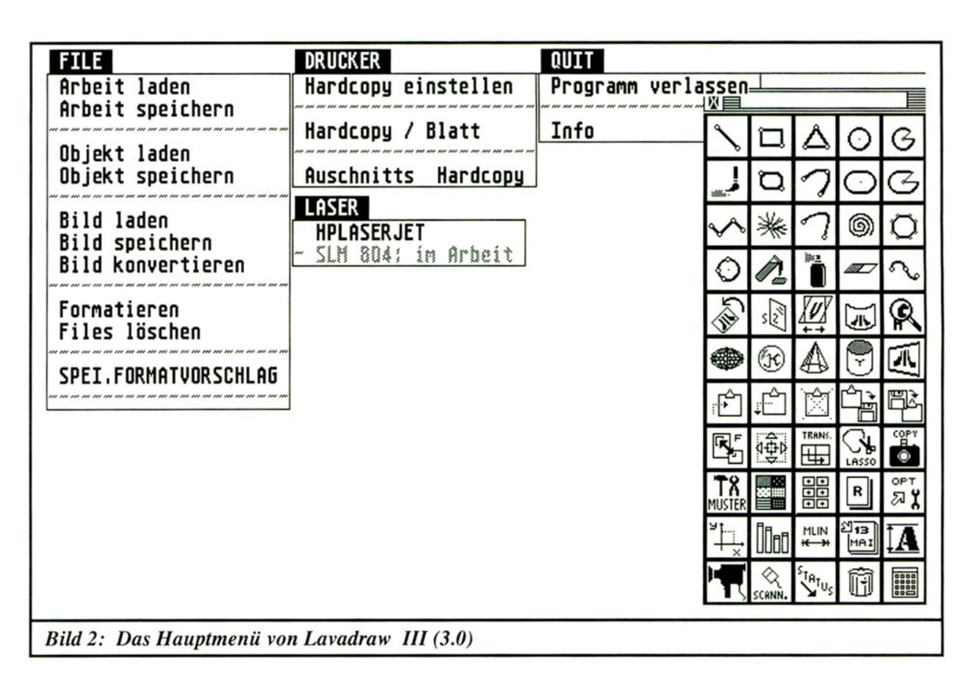


Bild 2: Das Hauptmenü von Lavadraw III (3.0)

Dem Zeichner stehen maximal acht mal zwei Bildschirme (also acht DIN A4-Seiten mit 640x800 Punkten) zur Verfügung, was im Normalfall ausreichen dürfte. Legt der Zeichner in erster Linie Wert auf ein professionelles Handbuch und ein durchdachtes Programm, das fast keine "Schnicki-Micki"-Funktionen enthält, dafür aber fehlerfrei scheint, so sind 99,- DM in dieses Programm gut investiert, wenn auch andere Programme für gleiches Geld zum Teil mehr Funktionen bieten.

LAVADRAW III (3.0)

Der Käufer fühlt sich wie "über den Wolken", wenn er die Vielzahl der Funktionen entdeckt, die ihm Lavadraw bietet. Ohne die Peripheriezugriffe stehen sage und schreibe 55 belegte Funktionssymbole zur Verfügung, die geduldig auf ihre Benutzung warten.

Dem Grafiker werden maximal 25 Bildschirme zu 640x384 Punkten (Screen abzüglich Menüleiste) zur Verfügung gestellt, wenn er dafür einen Mega mit mindestens zwei Megabyte als Gegenzug (sozusagen im Tausch) zur Verfügung stellt. Den Einmegabytelern bleiben immerhin noch neun Schirme.

Im GEM-Pull-Down-Menü lassen sich alle Funktionen abrufen, die sich auf die Außenwelt beziehen, auch das Löschen von Dateien und Formatieren von Disketten wurden nicht vergessen. Bilder und Objekte können geladen werden, wobei sich sowohl "STAD-gepackte" Bilder, GEM-Images und Screenformat als auch DEGAS-Bilder einlesen und auch speichern lassen. LAVADRAW packt sogar besser als STAD selbst, auch wenn pro Bild nur rund 200 Bytes "rausspringen" (diese Bilder verarbeitet das Original-STAD völlig korrekt). Eine gute Idee ist die Vorschlagsmöglichkeit des Speicherformates. Hierbei wird kalkuliert, wie lange IMG und STAD wären, und dem Benutzer das günstigere Format angezeigt.

Im Druckmenü kann der Benutzer zwischen 9-Nadel-, 24-Nadel- oder Laserdruckern wählen und verschiedene Druckgrößen und -auflösungen einstellen, in denen gedruckt werden soll. Die Bildschirme können sowohl längs als auch quer gedruckt werden. Weiterhin stehen acht Bilder pro DIN A4 zur Verfügung, die dann nahtlos aneinander gedruckt werden. Auf NEC P7- oder Epson LQ-1000-Druckern kann man sogar einen A2-Ausdruck realisieren. Der Ausdruck in höherer Qualität erfolgt um 90 Grad gedreht, wodurch sich endlos lange Bilderketten und Banner auf Rollen- oder Endlospapier realisieren lassen. Auch der Ausdruck eines Ausschnittes wurde nicht vergessen. Im Moment wird gerade eine

Anpassung an den ATARI SLM804-Laserdrucker entwickelt, HP-Laserjet-kompatible Laserprinter sind bereits integriert. Ferner ist die Ansteuerung des Printers/Flachbett-Scanners/Kopierers von HAWK in Vorbereitung. Der Handyscanner kann bereits mit der vorliegenden Version ebenso eingesetzt werden, um dem ST beliebige Grafiken "reinzuziehen", wie der Easytizer der MAXON-Computer GmbH.

An Zeichenfunktionen stehen neben dem alltäglichen Standard zahlreiche weitere Funktionen wie zum Beispiel Kreissegmente und Ellipsennetze zur Verfügung, ebenso ein Bogen durch drei Punkte oder ein Fächer. Aber LAVADRAW bietet auch viele Funktionen zur Bildverfremdung und -bearbeitung sowie Sonderfunktionen (siehe Hauptmenü in Bild 2). Hier finden sich für den Benutzer unter anderem eine Kurve durch n Punkte, das Projizieren auf Zylinder- oder Kugeloberfläche, ein Monatskalender (der nach eingegebenen Monats-/Jahresdaten berechnet wird), ein Mini-Taschenrechner und vieles mehr.

Die Lassofunktion ist allerdings nicht besonders gut geraten. Wer das Lasso von STAD kennt, wird zutiefst enttäuscht sein. Es wird bei LAVA zwar "ausgeschnitten", aber das war es dann auch schon. An eine Umrißberechnung (also ein Lasso, das sich zuzieht) ist nicht gedacht worden. Schade!

Die Anbindung an eine Festplatte ist ebenfalls noch nicht gelöst. Die Vorgabe des Laufwerkes und Ordners, von dem aus gestartet wurde, ist auf die Dauer lästig. Will man Bilder von Festplatte laden, so geht einem die ewige Ordner-"Klickeritis" genauso auf den Wecker wie die Fummelei, um die Laufwerkskennung zu ändern.

Bild 4: Die drei Menüs von GAMMA_03

Eine sehr schöne Möglichkeit bietet der mitgelieferte Zeichensatz-Editor (Bild 3), mit dem sich aus Bildern Rechtecke als Zeichen ausschneiden und so ganze Fonts erstellen bzw. aufgrund des mitgelieferten Bildschirm-Einfrierers SNAP auch aus Programmen "klauen" lassen. Es ist verwunderlich, daß am ST noch niemand vorher auf diese Idee gekommen ist. Bei Malprogrammen auf dem Commodore 64 ist dies bereits knapp drei Jahre möglich. Aber nicht nur zum "Diebstahl" wird Gelegenheit gegeben, mit Hilfe der Funktionen des Editors lassen sich auch (relativ) mühelos eigene kreative Ideen in schöne Fonts umsetzen, die dann leider im programmeigenen Format (oje!) gespeichert werden müssen. Nicht unerwähnt soll bleiben, daß der kostenlose Zeichensatzhelfer beim Einladen von Fonts während unseres Tests hin und wieder abstürzte. Naja, was soll's? "Einem geschenkten Gaul schaut man nicht ins Maul!" Schließlich ist dieser kleine Fehler besser als kein Editor...

Die UNDO-Funktion kann nur während des Zeichnens in einer Funktion wirksam gemacht werden. Verläßt man aus Versehen mit der rechten Maustaste diesen Modus, ist keinerlei Korrektur mehr möglich, was sehr gewöhnungsbedürftig ist und oft zum Verlust schöner Malergebnisse (man nenne nur Füllen) führen kann. Dieser Faktor ist leider ein entscheidendes Manko bei LAVADRAW. Auch ist das stetige Anwählen der Malfunktionen nach Verlassen des Bildes auf die Dauer mühsam und störend. Es wäre wünschenswert, den Modus beibehalten zu können.

Durch absolute und relative Koordinaten und das Einblenden von Linealen kann auch maßstabgetreues Arbeiten realisiert werden, ferner ist halbautomatisches Bemaßen möglich.

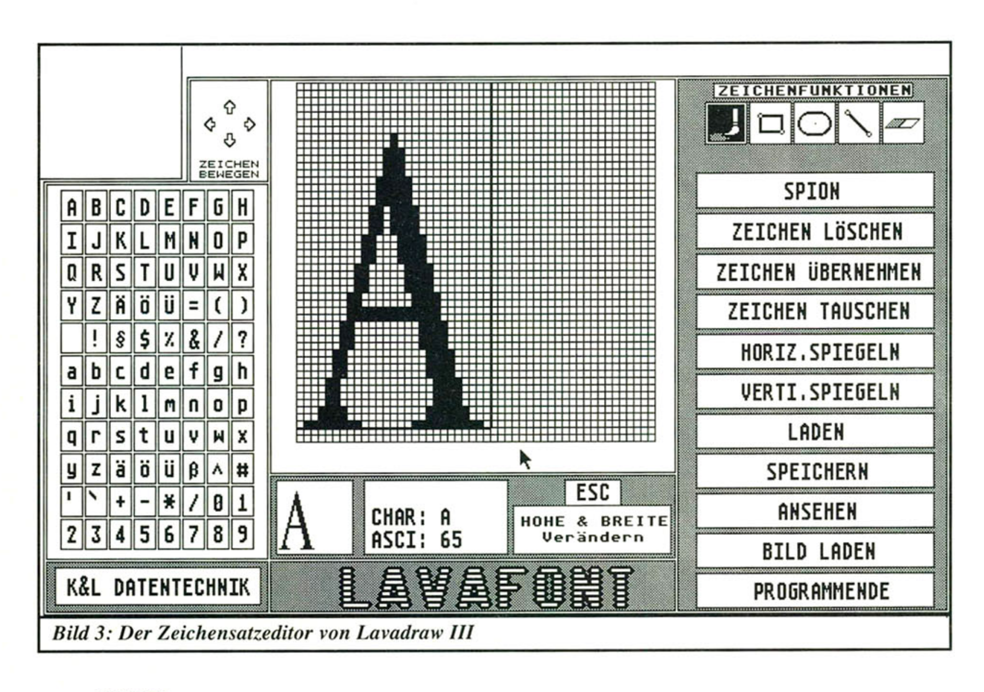


Bild 3: Der Zeichensatzeditor von Lavadraw III

Resümee

Das Programm erklärt sich weitgehend selbst und ist einfach zu bedienen. Einige Funktionen sind für den täglichen Zeichenbedarf eher unnütz, dafür sind sie jedoch für Maler, die besonderen Wert auf Dreidimensionalität und Effekte (wie zum Beispiel Kugelverformungen, Tonnen, Zerren und ähnliches) in ihren Bildern Wert legen, äußerst interessant. Mit seinen 55 verschiedenen Zeichenfunktio-

nen könnte man Lavadraw durchaus im gehobenen Zeichenbereich (kein CAD!) ansiedeln, was nur durch das Fehlen der oberen 16 Punkte (Menüleiste) stark beeinträchtigt wird. Somit wird das Bearbeiten mancher ganzseitigen Bilder aus anderen Programmen geringfügig eingeschränkt. Nützlich ist vor allem das Laden und Speichern von STAD-Bildern, was in manch anderem Zeichenprogramm fehlt. Zieht man den Preis in Betracht, so muß man sagen, daß LAVADRAW die Konkurrenten (auch Sympatic Paint) in bezug auf Anzahl der Funktionen schlägt und ein Preis von 119,- DM hierfür durchaus angemessen erscheint. Die Bedienung ist geringfügig gewöhnungsbedürftig, was jedoch bereits nach ein paar Stunden kein Problem mehr darstellt.

GAMMA_03

Gamma_03 wird auf zwei Disketten geliefert. Programm und Handbuch sind ebenfalls in einer Art Videokassettenhülle verpackt. In insgesamt drei Menüs, die dem Programm STAD nachempfunden wurden, bieten sich dem Benutzer zahlreiche Funktionen, die in Bild 4 aufgelistet sind. Daß die Standardfunktionen Rechteck, Kreis, Linie, Freihand und Polygon vorhanden sind, ist selbstverständlich. Das 50 Seiten umfassende Handbuch, das alle Funktionen ausführlich erklärt, enthält außerdem eine Kurzfassung für den Ungeduldigen, der sich gleich an seinen Rechner stürzen möchte, ohne lange das Handbuch durcharbeiten zu müssen. Das Programm ist einfach bedienbar, ohne Studium des Handbuches bleiben jedoch einige hilfreiche Funktionen unentdeckt. Hier wäre eine Tastaturtabelle im Handbuch brauchbar, die alle Belegungen kurz auflistet.

Die Anzahl der Bildschirme ist vom Speicher des ATARI STs abhängig. Auf einem 1040ST mit einem Megabyte gibt es maximal 14 Bildschirme (640x400), bei 2MB werden es 44 und bei 4MB müßten theoretisch 104 Schirme zur Verfügung stehen. Die ersten beiden sind immer zu einem verschiebbaren Ganzseiten-DIN A4-Schirm zusammengefaßt. Bei 512 KB bleibt nur der ganzseitige Screen übrig. Bilder können entweder komprimiert oder im Screen- bzw. DIN-Format abgespeichert werden. Ferner kann der Benutzer auch Objekte einzeln in Rechteckform ausschneiden und speichern bzw. später wieder einfügen, die sich leicht mit GFA-BASIC einladen lassen.

Bei den Zeichensätzen hat der Benutzer die Möglichkeit, zwischen den Systemfonts des Computers und eingeladenen Fonts in zwei Größen zu wählen. Die Fonts (leider wieder eigenen Formates) werden als Objektbox gespeichert, die sich jederzeit mit Lupe und allen anderen Funktionen behandeln läßt. Weiterhin haben die Autoren in das Programm eine Elektronikbibliothek eingebaut. Hier stehen in drei verschiedenen Untermenüs verschiedenste Bauteile zur Verfügung, die sich pixelgenau zu einem Schaltbild zusammenfügen lassen, was für Hobbyanwendungen durchaus genügt. Der Vorzug dieser drei Menüs besteht darin, daß sich die drei Leisten getrennt einladen lassen. Mitgeliefert werden hier verschiedene Zeichensätze, doch mit der Lupe des Programms kann der Benutzer auch seine eigenen Symbolbibliotheken mühelos erstellen. Somit läßt sich die Elektronikbibliothek für allerlei Zwecke brauchen und mißbrauchen (zum Beispiel Fonts, Minigrafiken, Elektroniksymbole, oft benötigte Symbole und vieles mehr).

Ein großes Manko ist die Druckroutine. Sie bietet zwar für 9- und 24-Nadeldrucker eine Anpassung mit Normal- und Doppeldruck, aber im Normalfall sollte man während des Ausdruckes wegen des hohen Zeitbedarfs (DIN-A4 Seite $\geq$ 8 Minuten) zum Kaffeetrinken gehen, um seinen Adrenalinspiegel zu schonen. An die Benutzung einer Festplatte (Ordner) mit verschiedensten Pfaden ist bei Version 3 endlich gedacht worden. Das ewige "Hin- und Hergeklicke" zwischen gewünschten Ordnern und Standardpfad entfällt und kostet somit weder Zeit noch Nerven.

Die Lupe des Programmes ist nach einiger Kritik überarbeitet worden. Sie ist jetzt relativ schnell und äußerst komfortabel. Neben einer Verschiebemöglichkeit, ohne ins Menü zu müssen, und Funktionen wie Linie und Füllen können auch Rechtecke und Kreise gezeichnet oder Ausschnitte invertiert werden. Daß nach dem Verlassen der Lupe die UNDO-Funktion zur Verfügung steht, muß wohl kaum erwähnt werden.

Überaus nützlich ist auch die Möglichkeit, sich den ganzen DIN A4-Schirm auf Tastendruck verkleinert anzeigen zu lassen. Einige Menüpunkte sind zusätzlich über eine Taste zu erreichen und müssen nicht per Maus angewählt werden, was in manchen Situationen sehr brauchbar ist. Auch kann der Benutzer mit einem kleinen Rechteck beliebige Muster ausschneiden und so mühelos selbst definieren.

Besonders gut gelungen sind die Zerr- und Verbiegefunktionen. Neben sonst üblichen Sinusverbiegungen und dem Klappen in X- bzw. Y-Richtung kann der "Bierfilz"-Modus angewählt werden, mit dem bestimmte Ausschnitte in Kreisform verbogen werden können. Sehr interessant scheint die "Tuben-Verzerrung" nach Bewegungen der Maus. Hier können die kuriosesten Effekte erzielt werden.

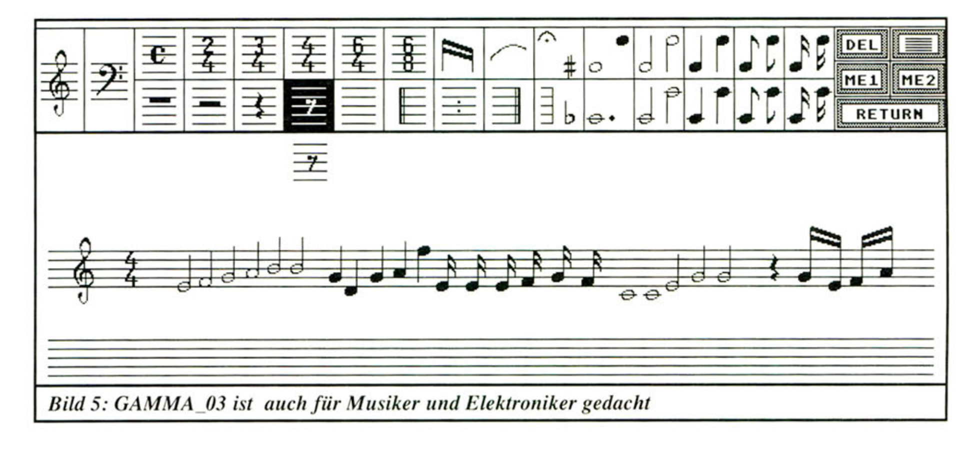


Bild 5: GAMMA_03 ist auch für Musiker und Elektroniker gedacht

Interessant sein dürfte GAMMA_03 auch für musisch Begabte: Ein gesonderter Menüpunkt läßt das Erstellen von ganzen Musikstücken auf Notenlinien (mit allen Notensorten, Pausen, Schlüsseln usw.) zum Kinderspiel werden (siehe Bild 5).

Die Ausführungsgeschwindigkeit mancher Routinen befindet sich eher im mittleren bis (nur teilweise) langsamen Bereich, aber dafür erhält der Anwender als Entschädigung viele interessante und nützliche Funktionen. Für den "Nicht-Freak" ist außerdem zu jedem Menü eine HELP-Leiste vorhanden, die die Symbole erklärt. Für den "Dreidimensionalisten" stehen Pyramiden, Zylinder, Dreiecke und Quader zur Verfügung.

Ein ganz besonderer Service ist das Programm "ALTERNATE", das kostenlos

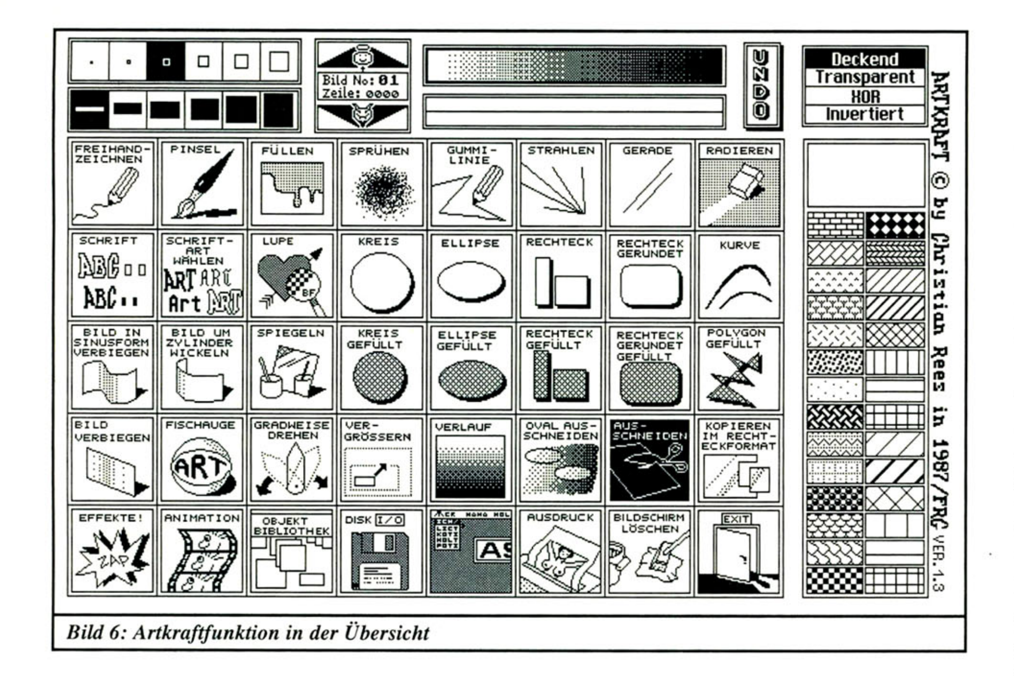


Bild 6: Artkraftfunktion in der Übersicht

zu Gamma_03 beigelegt wird. Hierbei handelt es sich um ein Programm, das Vorlagen vom Silverreed SPAT-Scanner einliest und in das gewünschte Bildformat umsetzt. Obwohl der Scanner normalerweise die Grafik entweder vergrößert oder verkleinert, hat der Programmierer mit Hilfe einiger Tricks eine Möglichkeit gefunden, eine DIN-A4 Seite genau auf 640x800 Punkte zu bringen. Dieser Scanner dürfte besonders interessant sein, da er auch als Fotokopierer verwendet werden kann und mit unter 2000,- DM preisgünstig ist. Ein genauerer Test wird in einer der nächsten Ausgaben folgen. Als Laserdruckerersatz und Fotokopierer kann man ihn auch deshalb sehen, weil zum Beispiel CALAMUS den SPAT mit seinen 200x200 dpi korrekt ansteuert.

Fazit

Das Programm ist mit zahlreichen Funktionen versehen. Vor allem die Zerrfunktionen und die Lupe verdienen ein großes Lob. Das Zeichnen geht locker von der Hand, auch ist das Vorhandensein bestimmter Funktionen in allen drei Menüs vorteilhaft, denn so entfällt das Umschalten dazwischen. Die äußerst langsamen Druckroutinen sollten jedoch für besseres Arbeiten unbedingt noch beseitigt werden. Aber auch so erscheint der Preis von 119,- DM mehr als nur angemessen. Gamma_03 bietet mit das beste Preis-/Leistungsverhältnis in unserem Test.

ARTKRAFT V1.31

Das Programmpaket Artkraft V1.31 ist zwar schon etwas älter - es wurde bereits 1987 erstellt - braucht aber deswegen seine aktuellen Konkurrenten keineswegs zu scheuen, sondern schlägt sie sogar in manchen Bereichen (zum Beispiel bei den Mustern) um Längen. Uns hat vor allem gefallen, daß sich das Programm weitgehend ohne Handbuch bedienen läßt. Dennoch ist so mancher Trick im Handbuch verborgen, das 40 Seiten umfaßt. Das Hauptmenü (siehe Bild 6) besteht aus vielen einzelnen Bildsymbolen (auch Icons genannt), insgesamt 40 an der Zahl. Neben den Standardfunktionen Linie, Kreis, Freihand, Rechteck usw. finden sich unter anderem Verbiegefunktionen (Sinus, Zylinder, Zerren, Fischauge), Effekte (Konturenvergrößerung, Solarisation) und viele andere.

Ferner kann der "Maler" ein GEM-Pull-Down-Menü aufrufen, in dem sich verschiedenste Parameter einstellen lassen. Interessant fürs Konvertieren einer Bildersammlung auf ein Format dürfte sicherlich die Vielzahl der einlesbaren Formate sein. Auch farbige Bilder lassen sich problemlos konvertieren, unbekannte Bilder (zum Beispiel aus Programmen, Spielen etc.) einlesen und so lange manipulieren, bis fast jede Datei brauchbare Ergebnisse liefert. Außerdem kann eingestellt werden, ob der Benutzer einen rechteckigen Block oder mit einem (echten) Lasso, das sich "zuzieht", ausschneiden möchte, ob Objekte wie Kreis und Rechteck einen Rand bekommen sollen usw. Ein äußerst guter Einfall war auch die Vordefinition von 16 statt 8 Graustufen. Hiermit lassen sich noch schönere Schattierungen und Grauverläufe erzeugen. Außerdem kann ein Lasso-Objekt mit einem ein- oder 16-farbigen Schatten unterlegt werden (siehe Bild 7).

Für Freihandzeichner dürfte die Interpolation eine nützliche Hilfe darstellen. Hierbei wird ein handgezeichneter Linienzug, der von Natur aus nie ganz zitterfrei ist, "geglättet". Natürlich kann diese Funktion ausgeschaltet werden. Zwar wird zum Programm ein Zeichensatzkonverter mitgeliefert, da leider auch Artkraft wieder einmal sein eigenes Fontformat

Bild 7: Menüleisten und Effekte von Artkraft

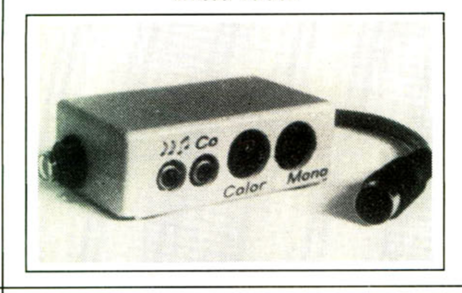
Auto-Monitor-Switchbox

Neu:
Mit automatischer Erkennung der Auflösung. Das Starten von Programmen in der falschen Auflösung gehört der Vergangenheit an, Programm anklicken: Programm wird automatisch in der richtigen Auflösung (Farbe oder Monochrom) gestartet.

Softwaremäßiges Umschalten o. Reset erstmals möglich **TASTATURRESET,** und Umschalten über die Tastatur.

Auto-Monitor-Switchbox

in neuer Version



Funktion Switchbox

Mit der **AUTO Monitor Switchbox** können Sie über die Tastatur aus jedem Programm heraus zwischen Monochrom und Farbmonitor umschalten oder einen Tastaurreset durchführen. Desweiteren ist es möglich über die Tastatur einen Kaltstart durchzuführen. Die mitgelieferte Software ist resetfest (arbeitet ständig im Hintergrund). Desweiteren ist es möglich durch Einbinden der von uns mitgelieferten Routinen **ohne RESET** zwischen Monochrom und Farbe umschalten.
Die Software wird nun in einer neuen Version ausgeliefert, die es ermöglicht, daß das Programm automatisch in der richtigen Auflösung gestartet wird. Das ständige Starten von Programmen in der falschen Auflösung gehört der Vergangenheit an.
Die AUTO Monitor-Switchbox verfügt zusätzlich über einen BAS und Audio Ausgang. Die Verarbeitungsqualität wird auch Sie überzeugen.
Update Software:
gegen Einsendung der Original Diskette + 15.00 DM.

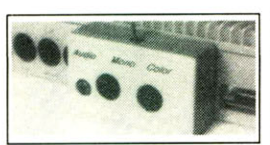
Monitor Switchbox

Monitor Switchbox, die sich nur über einen mechanischen Schalter zwischen Monochrom und Farbe umschalten läßt.

Monitorswitchbox mit 45 cm Kabel, zusätzlicher BAS und Audio Ausgang — **44.90**

Monitor Switchbox direkt an den Rechner anstöpselbar mit zusätzlichem Audioausgang — **39.90**

dto. als Bausatz (komplett) — **29.90**



Switchbox direkt an den Rechner anstöpselbar. (Alle Ausgänge sind weiterhin frei erreichbar. (s. Abb.) Jedoch nur für die Modelle 260/520/1040 ST ohne eingebauten Modulator geeignet.

Modul A.-M.-Switchbox

Mit diesem Modul stehen Ihnen alle Funktionen der AUTO Monitor Switchbox zu Verfügung.
Desweiteren verfügt dieses Modul über mehrere wichtige Tools die das Arbeiten mit dem Computer erleichtern, z.B. Fast Load, Mousebeschleuniger, Druckerspooler etc.
Das **Modul AUTO Monitorswitchbox** wird einfach in den ROM Port Ihres Rechners gesteckt. Die Garantie Ihres Computers bleibt erhalten. — **59.90**

Audiokabel Monitor Switchbox
5 polig, ca. 2 m — **7.90**

Audiokabel Monitor Switchbox
Cynch, ca. 2 m — **9.90**

Preise

Auto Monitor Switchbox
mit zusätzlichen BAS u. Audio Ausgang 45 cm Monitorkabel incl. Software — **59.90**

Auto Monitor Switchbox ST
direkt an den Rechner anstöpselbar mit zusätzlichem Audio Ausgang incl. Software — **54.90**
Wichtig: nur für die Rechnertypen 260/520/1040 ST ohne eingebauten HF Modulator geeignet.

Auto Monitor Switchbox Multisync
incl. Verbindungskabel Multisyncmonitor zusätzlicher Audio Ausgang, incl. Software
Wichtig: Monitortyp angeben — **69.90**

Diskettenlaufwerke

3,5-Zoll und 5,25-Zoll-Diskettenlaufwerke in vollendeter Qualität. Es werden nur die besten Materialien verwendet, z.B. Netzteil 5,25 Laufwerk vom größten europäischen Netzteilhersteller mit VDE, GS, Thermosicherung. Die Laufwerksgehäuse sind mit einer kratzfesten Speziallackierung ausgestattet. Die Chassis werden **nicht** über den Direktimport bezogen. Dieses ermöglicht eine ständige Lieferbereitschaft und einen guten Service.

Qualität und Service, der auch Sie überzeugen wird.

5,25 Disketten-Laufwerk



Anschlußfertiges 5,25 Zoll Diskettenlaufwerk (720 KB)
Laufwerkstyp **TEAC FD 55 FR**, robuste Verarbeitung. Unterstützt PC DITTO.
Aufpreis 2. Floppybuchse 27.90, Schaltung A/B 20.- — **339.-**

3,5 Disketten-Laufwerk



Anschlußfertiges 3,5-Zoll-Diskettenlaufwerk (720 KB)
Laufwerkstyp: **NEC FD 1037**
(25,4 mm hoch) robuste Verarbeitung, kratzfestes Gehäuse Netzteil VDE.
Einbau 2. Floppbuchse 27.90 — **249.-**

Software a la carte

Omikron Compiler	159.-	**Signum II**	357.-
		STAD	159.-

PC DITTO Version 3.64 — **169.-**

Public-domain (alle ST Disketten)
incl. Diskette eins. format. DM 5,90
Liste gegen DM 2,- in Briefmarken

PD Pakete 1 Paket = 5 Disk (doppelseitig) **35.-**
- **P.1** Die besten Spielprogramme
- **P.2** Die besten Utilitieprogramme
- **P.3** Die besten Mal/Soundprogramme
- **P.4** Die besten diversen Programme

Hardware Zubehör

Leerkarte Speichererweiterung
komplett bestückte Speichererweiterung (steckbar) ohne RAM's — **84.90**

Epromkarte
128 KB — **49.90**

Leerkarte Speichererweiterung
4 MB (Computertyp angeben) — **249.-**

Epromkarte 512 KB
incl. Treibersoftware zum Programmieren der Eproms — **119.-**

HF Modulator HF Modulator zum Anschluß des Atari ST an jeden gewöhnlichen Farbfernseher. Der HF Modulator zeichnet sich durch die besonders gute Bildwiedergabe aus. Der Ton wird über den Fernseher übertragen.
incl. FBAS, AUDIO Ausgang, Attenenkabel, hochwertiges Netzteil — **189.-**

Festplatten

Neu von Hard & Soft
professionelle Festplatten für den Atari ST.
- AUTO-Boot-fähig
- Durchgeführter DMA-Port
- schneller als die Atari Festplatte
- Festplattenkabel standardmäßig 2 m lang
- sehr leise (kein störender Lüfter)
- Abmessungen wie die orig. Atari Festplatte (gleichzeitig als Monitoruntersatz verwendbar)
- sehr robuste Verarbeitung — 1 Jahr Garantie

30 MB nur 1198.-
60 MB nur 1798.-

Fordern Sie unser kostenloses Zusatzinfo an.

abgesetzte Tastatur am ST

Abgesetzte Tastatur am ST
Tastaturgehäuse mit Spiralkabel, Treiberstufe Resettaste und Joystickportbuchsen eingebaut — **124.-**

ST Tast (XT Tastatur am ST)
Jede XT Tastatur am ST anschließbar, frei definierbare Tastaturbelegung, Tastaturreset — **139.-**

ST Tast + orig. Cherry Tastatur
Kombiangebot — **329.-**

Tastaturabdeckgehäuse Das Tastaturabdeckgehäuse wird einfach über den Atari ST gestülpt. Alle Ausgänge des Atari sind frei erreichbar. Durch diesen Zusatz wird der Computer zum idealen Monitorständer. Ideale Ergänzung zum Tastaturgehäuse und ST Tast — **59.90**

Hardware Zubehör

Uhrmodul intern gegenüber anderen Uhrmodulen benötigen Sie keine Software zum Anerkennen der Uhr. Die Bootsoftware befindet sich auf ROM's im Betriebssystem.
Wichtig: Betriebssystem angeben. ROM TOS o. Blitter TOS — **119.-**

Uhrmodul extern
incl. Treibersoftware — **89.-**

Mouse Pad rutschfeste Unterlage für Computermäuse (270 x 220 mm) — **17.90**

3 Laufwerke am ST

Floppyswitchbox: ermöglicht den Anschluß von drei Laufwerken am ST. Das Umschalten erfolgt ohne Reset. Die Switchbox ist mit speziellen Treibern für 3.5 u. 5.25 Laufwerke ausgestattet (m. Zugriffs-LED-Anzeige)
Wichtig: Computertyp angeben — **89.-**

Multisync Monitore

Monitor NEC Multisync GS — die echte Alternative zum SM 124
Auflösung 900 x 700, alle drei Auflösungen des Atari in SW darstellbar, bestechend scharfe Bildqualität, gebaut nach ergonomischen Richtlinien mit integriertem Schwenk-/Neigefuß
1 Jahr Garantie (keine Importware) — nur **579.-**

Monitor NEC Multisync II Auflösung: 800 x 560, alle drei Auflösungen vom Atari ST darstellbar, bestechend scharfe Bildwiedergabe, gebaut nach ergonomischen Richtlinien mit integrietem Schwenk-/Neigfuß
1 Jahr Garantie (keine Importware) — nur **1698.-**

Verbindungskabel

Floppyverlängerung (0,7 m)	29.90
Monitorverlängerung (1,5 m)	29.90
Harddiskkabel (1,5 m)	39.90
Druckerkabel (2 m)	24.90
Scartkabel (2 m)	39.90

Stecker

Monitorstecker	6.40
Monitorbuchse	8.90
Monitorkupplung	9.90
Floppystecker	7.90
Floppybuchse	9.90
Floppykupplung	9.90

Disketten

3,5 Fuji 2 DD	34.90
3,5 Fuji 2 DD (neutral verpackt)	27.90
5,25 Precision 2 DD (5,25 Zoll Disketten höchster Qualität)	10.90

Zubehör/Disketten-LW

Floppykabel Atari 3,5 Disketten-LW	27.90
Floppykabel Atari 5,25 Disketten-LW	29.90
Floppygehäuse NEC FD 1036, 1037 Teac FD 55 FR, mit Zugentlastung u. Gummifüße	29.-
NEC FD 1037 mit Anschlußbelegung	197.-
Teac FD 55 FR 229.- Atari modif.	239.-

Hard & Soft A. Herberg, Bahnhofstr. 289, 4620 Castrop-Rauxel, Tel. 0 23 05/1 57 64, Fax. 12022 — Händleranfragen erwünscht —

benötigt, dieser war jedoch in der vorliegenden Version insofern nicht besonders brauchbar, da nur MONOSTAR Plus-Fonts in das Artkraft-Format umgewandelt werden konnten. SIGNUM- und GEM-Zeichensätze sind angeblich jedoch in Vorbereitung.

Ein großes Manko ist es, daß der sonst ziemlich gute Druckertreiber bisher keine 24-Nadeldrucker unterstützt. Bei größerer Nachfrage soll dieser jedoch laut Auskunft des Programmierers nachgerüstet werden. Auch Animationen sind mit dem Programm in begrenztem Umfang möglich (bei weitem nicht so interessant wie bei Sympatic Paint). Das ist besonders interessant, weil dem Besitzer von 1MB zwar nur 15, dafür den MEGAs jedoch 47 (2MB) beziehungsweise 112 (DIN A5) Bildschirme (MEGA 4) zur Verfügung stehen.

Die erstellten Bilder werden ausschließlich im Degas-Format gespeichert, um anderen Programmen gegenüber Kompatibilität zu gewährleisten. Neben Bildschirmen können aber auch Objekte einzeln auf Diskette gesichert werden. Das Programm arbeitet problemlos mit der Festplatte und deren Ordnern zusammen.

überzeugen. Sicher wird sich für 24 Nadeln bei Nachfrage schnell eine Lösung mit dem Programmierer vereinbaren lassen. Nötig wäre vor allem noch eine Ausweitung der nutzbaren Zeichensätze, insbesondere eine Ausdehnung auf Fonts im GEM-Format, was aber sicher mit Erscheinen des GFA-BASIC-Compilers 3.0 kein großes Problem mehr darstellen dürfte. Zwar ist das Programm mit DM 119,- nicht gerade das billigste, dafür hat es uns aber recht gut gefallen, besonders wegen der guten Konzeption in bezug auf Anwahl der Funktionen, der konsequenten Einhaltung der Maustastenbelegung und der hohen Arbeitsgeschwindigkeit.

PUBLIC PAINTER

Dieses Programm wird in einem Ordner mit zwei Disketten und einem rund 70 Seiten umfassenden Handbuch geliefert. Startet der Benutzer das Programm, wird er zunächst mit fetziger, aber computerbedingt leicht "blechartiger" Musik begrüßt. Betrachtet der Zeichner danach seinen Public Painter genauer, der keineswegs Public Domain-Software ist, so stellt er fest, daß alle Pull-Down-Menüs in englischer Sprache sind (siehe Bild 8).

File	Block	Style	Extras	Options
New Picture	Rot left, right	Shade Black SF1	Undo	Set Font
Load *.*	Flip Mirror	Shade White SF2	✓ Block Zoom	Set Line
Load *.CMP	Half W Double W	Shade Light SF3	Erase All	Set Fill
Load *.IMG	Half H Double H	Outline SF4	Convert...	Set Bend
Abandon	Smooth Double WH	Light SF5	Copy to...	Set Spray
Read Text	Bending & Use	Bold SF6	Invert All	Set PRINT
Save *.*	Shadow	--TEXT-ONLY--	Fill Load	Set Look
Save *.CMP	Save as *.IMG	Center X SF7	Fill Catch	Set Snap
Save *.IMG	Save as *.ICN	Center Y SF8	Fill Save	Set Lasso
Erase *.*	Print Block	Justify Space SF9	View Full	Set Clock
Print Picture		Justify Chars AF9	View Layout	Set Pie
QUIT/Close		Calc Text SF10	View Clip	Set Bezier
		Home Text Home		Set f(x)
		Normalize Ins.		
		Use ⇧ ⇩ ⇦ ⇨ -Keys & SHIFT to move		

Bild 8: Die Menüleiste von Public Painter Monochrom (PPM) V0.45

Auch das Hin- und Herschalten zwischen mehreren Laufwerken (z.B. Diskette, RAM-Disk, verschiedene Festplattenpartitionen) ist kein Problem.

Zwar hat das Programm keinen eigenen Mustereditor, dafür bietet es jedoch anderweitig viele Funktionen, zum Beispiel Verlauf und Effekte. Daß das Programm zum Teil in Maschinensprache programmiert wurde, fällt bei einigen rechenintensiven Routinen durch angemessene Arbeitsgeschwindigkeit ins Gewicht. Die Bedienung ist einfach, das Konzept gut durchdacht. Vor allem die Vielfalt der Möglichkeiten und der schöne Ausdruck - wenn auch bisher nur auf 9-Nadlern -

Dies mag zwar besser klingen, aber für den "Nicht-Engländer" ist dann ein ständiges Handbuchwälzen notwendig, was durchaus lästig werden und die Mal-/Zeichenfreude verderben kann.

Das Programm läßt sich nach einer kurzen Einarbeitungszeit (fast) problemlos bedienen, wenn auch manche Funktionen etwas umständlich scheinen (ein Tester: "Wo ist denn hier der verflixte Radiergummi, muß ich denn alles mit der Lupe pixelweise löschen oder 'Blöcke vernichten'?"). Das Programm liest die meisten gängigen Bildformate bis auf STAD ein. Es hat einen eigenen Bildkompressor für das *.CMP-Format (siehe Bildkompaktor in ST-Computer 12/86), verarbeitet aber auch *.IMG-Files, wodurch eine Anbindung an Wordplus geschaffen wurde. Ferner können auch Farbbilder niedriger und mittlerer Auflösung eingelesen und konvertiert werden. Gespeichert werden kann im Screen-, IMG-, CMP-, Profi-Painter-, DIN A4- und Degas-Format, was sicher ausreicht.

Der Ausdruck kann in drei verschiedenen Druckdichten sowohl auf 9- als auch auf 24-Nadel-Druckern erfolgen. Die Parameter können vom Benutzer eingestellt werden, womit zum Beispiel ein gequetschter Ausdruck möglich ist, oder der exotischste Drucker angepaßt werden kann.

Die Lupe des Programmes arbeitet in vier verschiedenen Auflösungen über ein GEM-Fenster. Soll der Ausschnitt unter diesem Fenster ediert werden, muß erst das Fenster verschoben werden, was vor allem bei größeren Bildern lästig werden kann. Dafür wurde die Leiste der Zeichenfunktionen intelligenterweise zusätzlich in ein Pull-Down-Menü verlegt. Dadurch entfällt bei großen Schirmen das Verkleinern und Vergrößern des aktuellen Fensters. Neben den Standardfunktionen bietet PPM dem Benutzer einen hervorragenden Funktionenplotter an, über den sich gewiß Mathematiker und Physiker besonders freuen werden.

Auf die Zerr- und Biegemöglichkeiten wurde besonderer Wert gelegt. Vergrößern auf das Doppelte wurde ebenso wie stufenloses Ändern der Größe implementiert. Mit der Funktion "SMOOTH" kann nachträglich eine Glättung der groben Linien durchgeführt werden, wodurch die Auflösung nicht ganz verlorengeht.

Das Interessanteste dürften wahrscheinlich die mitgelieferten Zeichensätze im GEM-Format sein. Hier stehen für jede Gelegenheit Fonts in beliebiger Größe zur Verfügung, auf der gelieferten Diskette befinden sich über 60 (sechzig!) verschiedene Zeichensätze. Sicherlich wurde deshalb besonderer Wert auf die Textfunktionen gelegt. Nicht nur ASCII-Dateien lassen sich in lesbare Grafiken umsetzen, der Text kann auch zentriert, im Blocksatz ausgegeben oder bei überstehenden Buchstaben (z.B. W,A,T usw.) ineinander geschoben werden (Kerning) und vieles mehr. Die zahlreichen Möglichkeiten der Textmanipulation bieten Hobby-Handbuchautoren, die Text und Grafik kombinieren müssen, ohne sich ein DTP-

Programm zulegen zu wollen, genau die richtigen Funktionen. Durch Doppelklick können im Programm die meisten Parameter eingestellt werden (Füllmuster, Linienart usw.).

Fazit

Das Programm ist mit 79,- DM der preiswerteste Konkurrent. Die Benutzung des Programmes und die Fenstertechnik sind insgesamt etwas umständlich, und einige Funktionen mehr würden sicher nicht schaden. Die englischen Menüs sind auf die Dauer relativ störend. Auf der Textseite ist Public Painter den Mitbewerbern um eine Nasenspitze voraus und bietet gute Möglichkeiten für die private, halbprofessionelle Druckerei - wegen der vielen mitgelieferten Zeichensätze. Wer die vielen Zusatzfunktionen der anderen Programme nicht benötigt, erwirbt für wenig Geld ein leistungsfähiges Programm, das ideal als Ergänzung zu einem weiteren Zeichenprogramm dient, auch wegen des günstigen Preises.

Sympatic Paint V1.40

Das Programm wird mit einem 150-seitigen Handbuch in einem stabilen Plastikschuber geliefert. Bei einem Preis von 298,- DM ist es mit Abstand der teuerste Konkurrent in unserem Vergleichstest. Ob der Anwender für sein (vieles) Geld einen vernünftigen Gegenwert bekommt, werden wir noch sehen.

Beginnen wir mit dem Handbuch: Die einzelnen Funktionen werden sehr ausführlich beschrieben, was die Bedienung erleichtern soll. Der Einstieg in das Programm wird jedoch ohne vorheriges "Schmökern" nicht so leicht fallen wie bei anderen Programmen (zum Beispiel Gamma_03). Diejenigen, die das Handbuch vor der ersten Programmbenutzung nicht in einem Zug durchlesen, werden wohl zu Beginn etwas verwirrt. Im Handbuch hat man nämlich die Funktionen durchnumeriert bzw. "durchbuchstabiert". Da wird von F-Funktionen, T-Funktionen und ähnlichem geschrieben, weil man anfangs auf die Abbildung der Symbole verzichtet hat. Spätestens nach der dritten Funktion wird das mühsame Abzählen am Bildschirm mit dem Finger (8,9,A,B,...,T - endlich!) oder wahlweise Blättern im Handbuch lästig. Die Beurteilung des Konzepts der Zahlen- und Buchstabengliederung mag Geschmackssache sein, uns hat es jedenfalls nicht sonderlich überzeugt!

Genug gewettert, kommen wir nun zum Programm: Daß Standardfunktionen wie zum Beispiel Linie, Freihandzeichnen, Rechteck (eckig und abgerundet), Kreis- und Ellipsenausschnitte bzw. -bögen vorhanden sind, muß wohl kaum besonders hervorgehoben werden. Dafür lassen sich aber bei fast jeder der knapp 40 Funktionen erstaunlich viele Parameter und Attribute einstellen, so daß man wesentlich mehr als nur 40 Funktionen zur Verfügung hat (siehe Bild 7). Es kann sowohl mit absoluten Koordinaten als auch mit relativen gearbeitet werden, die nach Wunsch immer in der obersten Zeile angezeigt werden können. Diese Menüzeile zeigt auch den angewählten Befehl und ob sich hierbei Attribute verändern lassen. Sympatic Paint zeigt sich geradezu als Attribut-Jongleur und bietet eine Fülle von Einstellmöglichkeiten. Das fängt bei der Liniendicke an und endet beim Einschalten der Umrahmung von Kreisen, Rechtecken und so weiter.

Ein sympathischer Künstler ist Sympatic Paint auch bei Animationen. Scrolling (Verschieben) in alle Richtungen ist ebenso vorhanden wie verschiedene Überblendeffekte. Hier kann also der Hobby-Video-Filmer seine privaten ATARI ST-Filmchen drehen, ein eigenes Drehbuch schreiben und seine "Show" auch noch mit Musik untermalen, die mit dem Soundsampler eingelesen wurde. Für kleinere Kompositionen auf dem Musikchip des Rechners stehen weitere Programmiermöglichkeiten bereit. Mit der Programmierung der Animationen durch den Benutzer hätte man sich bei G DATA allerdings noch etwas mehr Mühe geben können, da die Befehlseingabe über eine Buchstabenfolge nicht gerade die einfachste ist (Beispiel einer Sequenz: "S :7 4 K4 (1)5" usw.). Aber das Programm soll ja nicht dem IMAGIC-Compiler Konkurrenz machen, sondern hauptsächlich als Mal- und Zeichenprogramm seinen Weg finden.

Wer andere Malprogramme (STAD, DEGAS, usw.) bereits gewohnt ist und immer noch gerne damit arbeitet, wird sich umstellen müssen. Die Bedienung des Programmes ist anfangs ziemlich ungewohnt, was sich nach längerer, intensiver Bedienung sicherlich ändert (bei uns waren zwei Wochen nicht lange genug). Dies fängt schon bei der Belegung der Maustasten an: mit der RECHTEN Taste wird etwas ausgeführt, mit der LINKEN nicht. Auch die Einstellung der Attribute über die ATTR-Funktion ist gewöhnungsbedürftig.

An Bildformaten liest das Zeichenprogramm STAD-gepackt, Screen- und Degas-Format genauso ein wie auch Doodle-Bilder (DIN A4). Beim Speichern kann der Benutzer zwischen Screen, Degas und GEM-Image (*.IMG) wählen. Damit dürfte der Grafiker für die wichtigsten Programme gerüstet sein. Sympatic Paint stellt dem eifrigen Zeichner übrigens bei einem Megabyte 18 Bilder zur Verfügung (640x384 Punkte - A5 ohne Menüzeile). Bei einem ATARI MEGA ST mit mehr Speicher kann sich die Zahl bis auf 99 erhöhen. Es wäre sicherlich besser, nur 95 Bilder zu bieten, dafür aber nicht bei jedem Bild die oberen 16 Punkte abzuschneiden, womit das gleiche Problem wie bei Lavadraw zu beklagen wäre (s.o.).

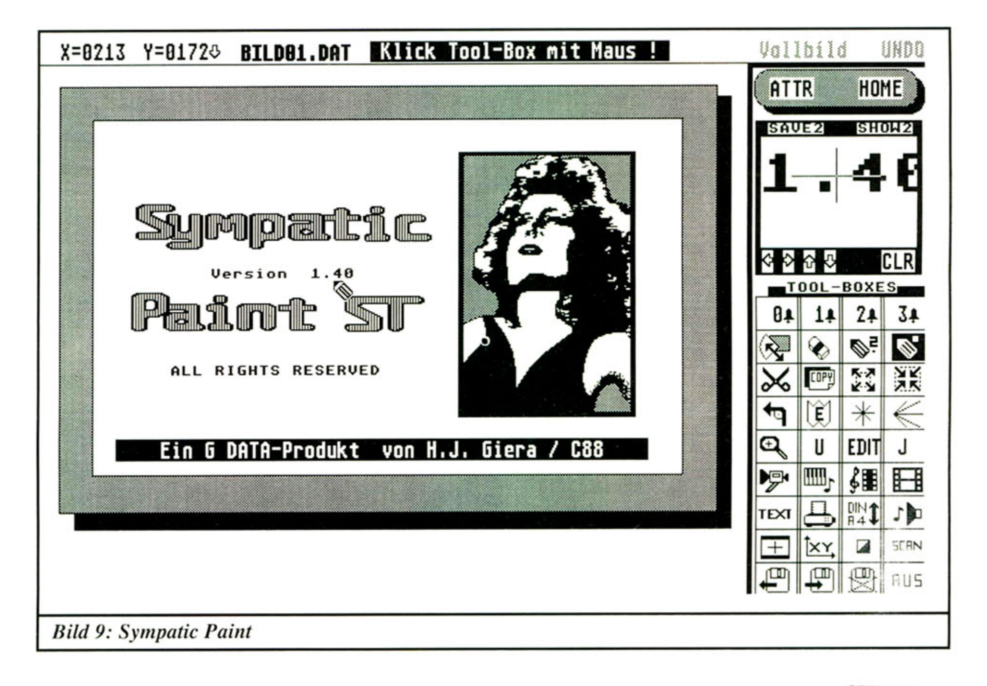
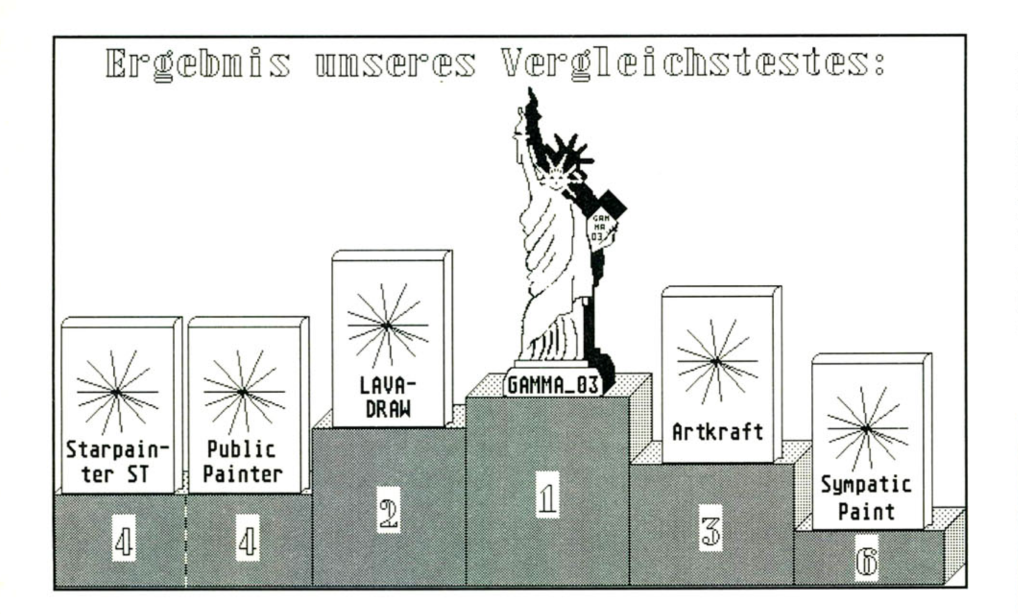


Bild 9: Sympatic Paint



In bezug auf die Bedienung war uns das Sympatic Paint eher unsympathisch, ganz im Gegensatz zu den reichhaltigen Manipulationsmöglichkeiten, die der Käufer jedoch sehr teuer zu bezahlen hat. Vergleicht man das Programm mit den Konkurrenten in unserem Test, scheint der Preis eindeutig überhöht, obwohl es auch Animations- und Sound-Möglichkeiten bietet (die zum Zeichnen wohl äußerst selten benötigt werden).

Allgemeines Fazit

Im Test bewegen sich alle Konkurrenten, bis auf das Produkt von G DATA, auf einem Preisniveau von 100,- DM. Die Leistungen sind effektiv in etwa gleich: Das eine Programm bietet ein paar Funktionen mehr, dafür arbeitet das andere wesentlich schneller. Am besten hat uns jedoch das Paket GAMMA_03 gefallen, das LAVADRAW um eine Nasenlänge schlägt, vor allem durch die bessere Bedienung, die UNDO-Funktion und die Festplattenanbindung. Zwar liefert LAVA den Zeichensatzeditor kostenlos mit, dafür kann dies bei GAMMA mit der Lupe verwirklicht werden, und zusätzlich stehen alle Zeichenfunktionen zur Verfügung. Außerdem haben Musiker und Hobbyelektroniker sicher an GAMMA_03 ihre wahre Freude. STARPAINTER ST ist sozusagen als Standardwerk zu betrachten, wird jedoch von ARTKRAFT eindeutig überboten, das vor allem in bezug auf die Funktionsvielfalt den 3. Platz verdient hat. PUBLIC PAINTER ist zwar ebenfalls preisgünstig, doch fällt dieses Programm wegen der mühsameren Bedienung gemeinsam mit STARPAINTER auf einen vierten Platz. Zeichensatzfreaks, die eine reichhaltige Palette an Fonts schätzen, werden sich sicher für PUBLIC PAINTER entscheiden, allein wegen der vielen Zeichensätze. SYMPATIC PAINT muß aufgrund seines äußerst hohen, aber unangemessenen Preises den sechsten und somit letzten Platz hinnehmen, da es trotz seiner Qualitäten das schlechteste Preis-/Leistungsverhältnis bietet.

Abschließend ist festzustellen, daß alle Programme uneingeschränkt brauchbar sind und ein zeitgemäßes, hohes Niveau aufweisen. Letztendlich liegt es daher am Kunden, welchem Programm er - unter Berücksichtigung seiner persönlichen Interessen - den Vorzug gibt. Es ist bedauerlich, daß einer weiten Verbreitung des SYMPATIC PAINT sein hoher Preis im Wege steht.

RP

Bezugsadressen:

GAMMA_03 [DM 119,-]
Computer-Center,
Turmstraße 16,
6730 Neustadt/Weinstraße
Tel.: 06321/32132

LAVADRAW III (3.0) [DM 99,-]
K&L Datentechnik,
Bahnhofstr. 11,
3551 Bad Endbach
Tel.: 02776/202 (ab 15 Uhr)

ARTKRAFT (V1.31) [DM 119,-]
Samsa-Soft,
Postfach 16 61,
2240 Heide/Holstein
Tel.: 0481/63088

PUBLIC PAINTER MONOCHROM (V0.45) [DM 79,-]
Axel Braukmann,
Am Sportplatz 51,
4005 Meerbusch 2
Tel.: 02159/1899

STARPAINTER ST [DM 99,-]
Sybex-Verlag GmbH,
Postfach 30 09 61,
4000 Düsseldorf
Tel.: 0211/61802-0

SYMPATIC PAINT (V1.40) [DM 298,-]
G DATA,
Siemensstraße 16,
4630 Bochum 1
Tel.: 0234/435553

ENDE

ANTI VIREN KIT

Unser neues Anti Viren Kit: erkennt alle bekannten Viren, erkennt ähnliche Viren — vernichtet alle bekannten Viren und deren Verwandte — erkennt bei vielen kommerziellen Programmen, ob gesund oder infiziert. — Datenrestbestände werden gerettet — immunisiert Disketten gegen Bootsektorviren. Die Disketten werden geimpft — repariert defekte und infizierte Bootsektoren — Link Viren werden von den infizierten Programmen abgeschnitten und vernichtet — intelligente Bootsektorscannung: erkennt auch noch unbekannte Viren. — Updateservice und Datenrettungsservice für registrierte Benutzer — auch für Harddisk **DM 99,—***

CHESS

Das stärkste Schachprogramm für ST Computer (bei 66.000 Eröffnungszügen) bietet nicht nur Spielstärke, sondern auch Komfort. Nachladbare und erweiterbare Eröffnungsbibliothek, beliebige Bedenkzeit (bei Ausnutzung Ihrer Bedenkzeit), Zugvorschlag, Seitenwechsel, Hängespiel, Problemschach, Patreindruck und Speicherung, Blitzen usw. sprechen klar für CHESS. **DM 129,—***

DISK HELP

Repariert Disketten und erlaubt in 99% der Fälle eine Wiederbelebung des gesamten Datenbestandes. Das Programm gehört neben jeden ST. Lassen Sie Ihre wichtigsten Daten nicht einfach von Laufwerken zerstören. DISK HELP ist einfach zu bedienen, korrigiert Lesefehler und rekonstruiert Files. Physikalische Fehler (Risse, Kaffee) können nicht behoben werden.

DM 79,—*

FAST SPEEDER

Für alle, die ihrem Massenspeicher keine Pause gönnen wollen und lange Wartezeiten satt haben. Bringen Sie Ihre Festplatte und Diskettenstation auf Trab! Im Extremfall erreichen Sie dabei Ramdiskgeschwindigkeit! Festplatten werden bis zu 200% beschleunigt. Diskettenlaufwerke bis zu 900%. FAST SPEEDER ist einfach zu handhaben. Der optimierte Spezialalgorithmus mit Cache Prinzip arbeitet für Sie unmerklich im Hintergrund.

DM 129,—*

G RAMDISK II+

In der neuesten Version mit allen Leistungsmerkmalen, die man sich für eine Ramdisk überhaupt wünschen kann: Resetfest bis 4 MB, deabschaltbar: integrierter Druckerspooler, Bootcopy; Parameter save usw. Die G RAMDISK II ist im harten Einsatz bereits tausendfach bewährt.

DM 49,—*

GEM Retrace Recorder

Der Knüller! Wie von Geisterhand bewegt sich die Maus, laufen Kommandos ab, werden Aktionen gestartet. Der GEM Retrace Recorder ermöglicht es, jegliche Aktion (Mausbewegung, Menüs, Klicks, Tastatur usw.) aufzuzeichnen, und auf Tastendruck jederzeit original zu wiederholen. Das heißt: GEM Makros erstellen für Aktionen, die ständig wiederholt werden (auch in jedem Programm); super Auto Boots erstellen; selbstlaufende Demos von jedem Programm ohne Programmeingriffe und Kenntnisse u.v.m. Die Anwendungsmöglichkeiten sind fast unbegrenzt!

DM 99,—*

Harddisk Help & extension

Wir wollen keinesfalls die Ängste um Ihre täglichen Daten schüren, aber sind Ihre Daten wirklich sicher? Wie schnell führt ein Headcrash zum Ende! Sie sollten auf ein bewährtes Backupsystem nicht länger verzichten. Auch sehr große Files können einzeln gesichert werden. Partition Backup, Tree Funktion, selected Backup und Diskoptimizer erhöhen die Effizienz dieses Programms. Auch bei der Geschwindigkeit wurden keine Kompromisse geschlossen.

DM 129,—*

NEUHEITEN

SYMPATIC PAINT

neue Version!

Der ATARI ST als Grafik Workstation der Superlative! Alle Funktionen, die für ein effektives und komfortables Zeichnen, Malen und Konstruieren erforderlich sind, wurden integriert. Der Benutzer hat nicht nur die Möglichkeit der Erstellung einer individuellen Grafiktoolbox, sondern auch eine Animationsselektion, mit der Bilder laufen lernen. Bilder können dabei mit dem G SCANNER und EASYTIZER eingelesen werden.

Neue Version mit über 40 zusätzlichen Funktionen!

Preissenkung: **DM 199,—***
Handbuch vorab: **DM 30,—***

GRAPHIC WORKS

Als optimale Ergänzung zu dBase kompatiblen Datenbanken, VIP usw., erstellt GRAPHIC WORKS aus tabellarischen Zahlendarstellungen eindrucksvolle Präsentationsgrafiken. Die zahlreichen Grafiken (Balken-, Torten-, Liniengrafiken usw.) ermöglichen eine anschauliche und übersichtliche Darstellung von Zahlenkolonnen.
Interne Rechenblatt- und Makrofunktionen, automatische Regressionsanalysen mit freier Einteilung, Diashow für Vorträge, sind nur einige Features, die GRAPHIC WORKS zu einem extrem vielseitigen Arbeitswerkzeug machen.

DM 199,—*
Handbuch vorab: **DM 30,—***

G COPY II

neue Version!

Das bewährte vielseitige Kopierprogramm wurde mit einer neuen, stark verbesserten Oberfläche versehen, und mit zahlreichen Verbesserungen sowie neuen Features auf den Level von 1989 gebracht.
Mit seinen zahlreichen Zusatzplänen kann G COPY II nun noch mehr: kopiert und formatiert alle Formate (912 K-, 862 K-, 416 K-Disketten mit 10 oder 11 Sektoren und 80 — 83 Tracks) — optimale Kopiergeschwindigkeit bei 100% Datensicherheit — alle Programme ohne Hardwarekopierschutz werden kopiert. Neu: bis Track 90.
G COPY dient nur zur Erstellung von erlaubten Sicherheitskopien.
Raubkopien sind strafbar!
Für alle ST mit Monochrom- und Farbmonitor.

DM 99,—*

Hardware Uhr

Die Uhr wird ohne Lötarbeiten im Rechner eingebaut, so daß der ROM Port frei bleibt. Sie haben ständig die aktuelle Zeit und das aktuelle Datum! Die Batterie garantiert hohe Ganggenauigkeit und mindestens 3 Jahre Laufzeit bei voller Schaltjahrerkennung. Der sensationelle Preis für die anschlußfertige Uhr beträgt nur **DM 79,—***

INTERPRINT II

Die universelle Anpassung für jeden Drucker hat einen integrierten Treiber, der auch alle Umlaute und Sonderzeichen nach Ihren Wünschen konvertiert. Dabei können mit Hilfe des Editors maßstabsgerechte Hardcopies auf allen Matrixdruckern und HP-Lasern erstellt werden. Auch ein Druckerspooler und unsere G RAMDISK II+ sind integriert. Die Parameter sind speicherbar, und der komfortable Editor ermöglicht schnellste Anpassung.

DM 99,—*
ohne Ramdisk **DM 49,—***

M.A.R.S.

Core Wars (Krieg der Kerne, Bericht in ST 11/87): „Die neue Spielidee aus den USA. Für Strategen, die von Ballerspielen die Nase voll haben!"
Selbst programmierbare Viren kämpfen in Ihrem ST gegeneinander.
— grafische Kampfausgabe
— Trace Modus
— integrierter Editor
— voller Weltmeisterschaftsstandard
— ausführliche deutsche Dokumentation **DM 129,—***

RELAS

Aktienanalyse für den privaten und professionellen Aktienanleger auf der Basis der Relativen Stärke und Volatität von Charts wahlweise über Bildschirm und Drucker – Mischen von Charts einer Aktie und Branche zum Vergleich – Kurseingabe nur einmal wöchentlich (!) Als Basis dienen die letzten 27 Wochenschlußkurse – Umfangreiche Bereinigungsfunktion bei Kapitalveränderungen von Aktien - Listen der Relativen Stärke mit Mittelwert von Kursen und Analyse der Veränderung der Rangfolge der Relativen Stärke über 3 Wochen (wahlweise Bildschirm oder Drucker) – u.v.m.

DM 398,—*
Handbuch vorab: **DM 40,—***

Roboterarm

Der semiprofessionelle 5achsige Roboterarm wird anschlußfertig für ST, AMIGA und PC/AT geliefert. Für Schulungszwecke, Lehr- und Demonstrationszwecke oder einfach zum Spielen bestens geeignet. Technische Daten: Höhe ca. 54 cm, industriegelb, ca. 2,5 kg schwer, einfacher Anschluß an Druckerport. Auch über Joysticks zu steuern.

Anschlußfertig: **DM 269,—***
Spezialnetzteil: **DM 99,—***
Batteriesatz: **DM 10,—***

TV Modulatoren

Schließen Sie Ihren ST an den Farbfernseher an und sparen Sie sich einen teuren RGB Monitor. Unsere Modulatoren bringen ein scharfes Bild mit allen Mischtönen und einen Tonausgang extern und einen über den Fernseher. Das heißt: bestes Bild und bester Ton für alle Fernseher.

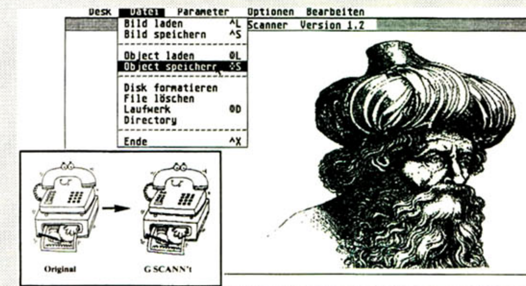
Preissenkung: Typ A (ATARI an TV) mit Monitorumschalter **DM 225,—***
Typ C FBAS Wandler für Geräte mit AV oder FBAS Eingang **DM 165,—***

AS SOUND SAMPLER

jetzt lieferbar:
AS SOUND SAMPLER III
Sampling in CD-Qualität

Schon der Sound Sampler II überzeugte Kunden wie Presse (Keyboards 12/87: „Verglichen mit 10-15-fach teureren Samplern ... muß man von einer guten bis sehr guten Qualität sprechen." ST 1/88: „Der Alleskönner!") Alle Leistungsmerkmale des Sampler III bietet der neue Sound Sampler III in 16 BIT CD Qualität! Aufnahmen, editieren, verknüpfen, mischen, transponieren, Echoeffekte, einbinden in Programme, Sounds oder MIDI Keyboards spielen; alles in original CD Qualität! Jetzt in 3 Versionen:

AS SOUND SAMPLER II Standard:	**DM 198,—***
AS SOUND SAMPLER II maxi:	**DM 298,—***
AS SOUND SAMPLER III 16 BIT:	**DM 598,—***
Soundbibliothek 10 Disketten für Sampler II	**DM 149,—***
Soundbibliothek 10 Disketten 16 BIT	**DM 198,—***



G SCANNER

Problemloses Digitalisieren von s/w-Bildern mit Ihrem Drucker ermöglicht der G SCANNER. Die hochwertige Abtastdiode von HP garantiert die ausgezeichnete Qualität von 200 dpi. Sie wird einfach auf dem Druckkopf Ihres Druckers befestigt. Mitgeliefert wird Software, die DEGAS und Normalbilder erstellt und Bildeditieren ermöglicht. Der G SCANNER ist voll kompatibel zu STAD und SYMPATIC PAINT (Scannoption).

s/w SCANNER:	**DM 298,—***
Diodenhalterung für NEC P6	**DM 35,—***
Graustufenscanner verwandelt Farbbilder zu Graustufen, voll kompatibel	**DM 398,—***

Vertrauen Sie auf 8 Jahre Erfahrung in Software.

Wir von G DATA bürgen durch langjährige Erfahrung für höchste Qualität und Effizienz.

Allen Computerfreunden wünschen wir ein frohes Weihnachtsfest und ein glückliches neues Jahr

Ihr G DATA Team

G DATA

Siemensstraße 16 • 4630 Bochum
Verkaufsbüro: Hattinger Str. 312,
4630 Bochum 1
Telefonische Bestellungen:
02 34 / 43 55 53

Schweiz: DATA Trade
Langstraße 94
CH-8021 Zürich
Österreich: Computershop
Rittner • Hauptstraße 34
A-7000 Eisenstadt

*unverbindliche Preisempfehlung
Irrtum und Änderung vorbehalten

Coupon

Hiermit bestelle ich

❑ Informationsmaterial ______
(DM 2,— beiliegend)

❑ 2 Demodisketten ______
und Infomaterial
(DM 10,— beiliegend)

❑ per Nachnahme ______
(ca. DM 6,50
Versandkosten) ______

❑ Verrechnungsscheck ______
liegt bei (+ DM 5,—
Versandkosten) ______

Absender:

______
______
______
______

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

1000 Berlin



DATAPLAY

Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61



Computare

Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d





Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

1000 Berlin



alpha computers g.m.b.h.
u. a. alphatronic, atari, commodore, dai, epson, sord mit pips, nec
hard-/software nach maß — servicetechnik
Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/891 1082

Computershop
Behrendt, Reinecke, Tscheuschner GbR
Riesen Software-Angebot
Fürbringerstraße 26 · 1000 Berlin 61
Tel. (0 30) 6 91 76 66 · BTX (030) 6 91 76 66

COMPUTER-STUDIO
Schlichting
... die etwas andere Computerei

Ihr Spezialist in Berlin für Hardware + Zubehör Eigenes Softwarestudio über 1000 verschiedene Titel am Lager

ATARI-Fachmarkt
NEC-Fachhandel · MS-DOS Fachmarkt

Katzbachstraße 6 + 8 · 1000 Berlin 61
☎ 030/786 43 40

2000 Hamburg

Computer & Zubehör-Shop
Gerhard u. Bernd Waller GbR
Kieler Straße 623
2000 Hamburg 54
☎ 040/570 60 07
BTX 040 570 52 75

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 44 00

Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 040/641 50 91

RADIX Bürotechnik
Heinrich-Barth-Straße 13
2000 Hamburg 13
Telefon (0 40) 44 16 95
NEU: Software Shop

Hardware
Software
Beratung
Service



ATARI Systemfachhändler
Münsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1



2000 Norderstedt



Ulzburger Str. 2
2000 Norderstedt
Tel. 0 40 / 527 30 40

2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt

Heiligengeiststr. 20, 2120 Lüneburg
Tel. 0 41 31 / 4 61 22, Btx 40 24 22
Mo.-Fr. 9°°-18°° und Sa. 9°°-13°°

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Bübel Martin Kopplow

Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

4150 Krefeld



- Festplatten
- Scanner
- Drucker
- BTX-Module
- Literatut
- Zubehör

COP Computer Service GmbH
Lewerentz-Straße 111 · 4150 Krefeld
Telefon (0 21 51) 77 30 42

4250 Bottrop

Megateam-Computer-Systeme

Kirchhellenerstraße 262
4250 Bottrop

4300 Essen

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4320 Hattingen



Hattingen, Große Weilstr. 18-20, Telefon (0 23 24) 20 94 77

4330 Mülheim

BNS Computer und Bürotechnik Vertriebsgesellschaft mbH
Dickswall 79 4330 Mülheim Telefon 02 08/3 40 34



Computer Hard- und Software auch im Leasing
Computerkurse für Anfänger und Fortgeschrittene

VICTOR SEL-Fernkopierer
NEC ATARI OKI

4350 Recklinghausen

Handwerker & Selbstständige
Ihr Computerpartner
Computer kauft man da, wo
der Service stimmt!
EDV-THIEL 02361/651490

4422 Ahaus

ATARI · Epson · Fujitsu Molecular · NCR · Tandon · Schneider · Star

OCB

OCB-Computershop Wallstraße 3 4422 Ahaus Tel. 0 25 61/50 21	OCB-Hard- und Software Wessumerstraße 49 4422 Ahaus Tel. 0 25 61/50 21

4430 Steinfurt

CBS GmbH
COMPUTERSYSTEME

Tecklenburger Str. 27
4430 Steinfurt-Burgsteinfurt
☎ 02551/2555

4500 Osnabrück

Heinicke-Electronic

Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41-8 27 99

Wir liefern Micro-Computer seit 1978

4600 Dortmund

ATARI SYSTEM-Fachhändler



BÜRO STUDIO BOLZ

4600 Dortmund 1 · Brauhausstraße 4
Telefon (02 31) 52 77 13-16

Elektronik
Computer
Fachliteratur

ATARI-System-Fachhändler

4600 Dortmund 1, Güntherstraße 75, Tel. (02 31) 57 22 84



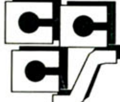
city-elektronik

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Kampstraße 1 · 4600 Dortmund
Telefon (02 31) 5 43 91

cc Computer Studio GmbH



Atari-Systemfachhändler
PCs von Tandy Schneider Peacock
Drucker von Star Brother NEC

Elisabethstr. 5
4600 Dortmund 1
Tel. 0231/528184 Tx 822631 cccsd Fax 0231/528131

4620 Castrop-Rauxel

R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 · ☎ (0 23 05) 3 77 0 · 4620 CASTROP-RAUXEL

ATARI
System-Fachhändler

4650 Gelsenkirchen-Horst

MENTIS GmbH

Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

4700 Hamm

computer center



4708 Kamen



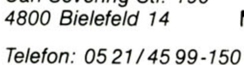
D+M Computer
Weststraße 45
4708 KAMEN
02307-17052

BTX 02307 15181 · Telefax 02307/1 83 67

4800 Bielefeld



CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106-108
4800 Bielefeld 1
Tel. (05 21) 6 16 63



Carl-Severing-Str. 190
4800 Bielefeld 14

Telefon: 05 21/45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21/45 99-123

MICROTEC
Software
Hardware
Beratung
Service

4950 Minden



- Software
- Hardware
- Organisation
- Beratung
- Schulung

Obermarktstr. 21
4950 Minden
0571 - 2 14 48

5000 Köln



AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
TEL. (02 21) 21 91 71

6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing

Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6500 Mainz

:ELPHOTEC
Computer Systeme
Ihr Atari Systemhändler mit eigenem Service-Center
Walpodenstraße 10
6500 Mainz
Telefon 0 61 31 - 23 19 47

6520 Worms



Computersysteme
GmbH

6520 Worms · Friedrichstraße 22
Telefon 0 62 41 / 67 57 - 58

6700 Ludwigshafen

MKV Computermarkt

Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

THEILLE Computersysteme
Gilgenstraße 4 · 6720 Speyer
Telefon (0 62 32) 772 16

6800 Mannheim



Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912

Computer-Center
am Hauptbahnhof GmbH

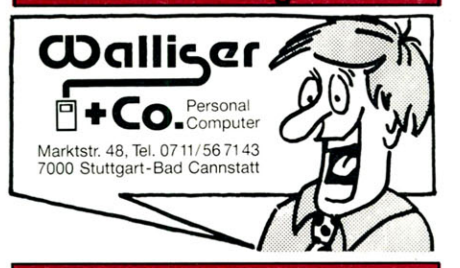
L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83 / 84

6900 Heidelberg

JACOM FAMILA-CENTER
Hardware · Software
Schulung · Service
Hertzstraße 1 · 6900 Heidelberg 1
Telefon (0 62 21) 30 24 37

HEIDELBERGER COMPUTER CENTER
Bahnhofstraße 1
6900 Heidelberg
Telefon 06221/27132

7000 Stuttgart



7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler
ATARI ST
Matrai computer
Matrai Computer GmbH
Bernhauser Str. 8
7022 L.-Echterdingen
☎ (07 11) 79 70 49

7030 Böblingen

Verkauf – Service – Software

Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 0 70 31 / 22 60 15



COMPUTER CENTER

7047 Jettingen

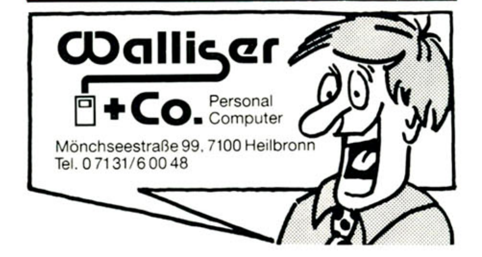
Verkauf – Service – Software

Norbert Hlawinka
Heilbergstraße 3
Im Multi-Center
7047 Jettingen
Telefon (0 74 52) 7 76 15



COMPUTER SHOP

7100 Heilbronn



7100 Heilbronn



Computer-Welt
Am Wollhaus 6
7100 Heilbronn
Tel. 0 71 31 - 6 84 01 - 02

7101 Flein

der **COMPUTERLADEN** von


In der Falterstraße
7101 Flein
Beratung, Service, Verkauf,
Software-Entwicklung
direkt beim Peripherie-Hersteller

7150 Backnang



7400 Tübingen

Werner Brock
COMPUTERSTUDIO
Poststraße 2—4 · D-7400 Tübingen
Tel. (0 70 71) 3 43 48 · Fax (0 70 71) 3 47 92
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR...

7410 Reutlingen

Werner Brock
COMPUTERSHOP
Federnseestr. 17 · 7410 Reutlingen
Tel. 0 71 21 - 3 42 87
Tx 1 72 414 024 RMI D · box rmi taisoft · Fax 0 71 21 - 33 97 79
Autorisierter Systemfachhändler für:
ATARI, Schneider, Commodore, Panasonic, Kaypro, Sharp, NEC, OKI, STAR,...

7450 Hechtingen



Gesellschaft für Datenverarbeitung mbH

Computer · Drucker
Zubehör · Fachliteratur
Schloßplatz 3 · 7450 Hechingen
Telefon 0 74 71 / 1 45 07

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT
HARDWARE · SOFTWARE · LITERATUR


ATARI COMMODORE CUMANA DATA-BECKER MULTITECH RITEMAN SCHNEIDER THOMSON

7475 Meßstetten 1 · Hauptstraße 10 · 0 74 31 / 6 12 80

7480 Sigmaringen

Wir stellen uns für Sie auf den Kopf!

Wir führen:
Hardware - Software - Zubehör
Zeitschriften - Bücher

Wir sind autorisierte Fachhändler von:
ATARI · Schneider · Tandon

SOFT & EASY COMPUTER GMBH
7480 Sigmaringen, Am Rappgassle, Tel. 07571/12483
7968 Saulgau, Pfarrstraße 13, Tel. 07581/2598

7500 Karlsruhe

ERHARDT Am Ludwigsplatz
Am Ludwigsplatz · 7500 Karlsruhe 1 · Tel. (0721) 1608-0

MKV GMBH
Kriegsstraße 77
7500 Karlsruhe
Telefon (0721) 84613

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk
In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7640 Kehl/Rhein



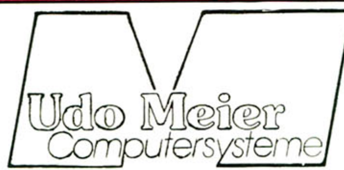
Computer · Software · Marketing
eigener Service · eigene Software

Badstrasse 12
Tel. 07807/822
Telex: 752913
7607 NEURIED 2

Filiale:
Hauptstrasse 44
Tel. 07851/1822
7640 KEHL/RHEIN

ELEKTRO-MÜNTZER GmbH

7700 Singen



Ringstraße 4
Telefon (07731) 68222

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER
computer - fachgeschäft
rösler
Rheingutstr. 1 · ☏ 07531-21832

7800 Freiburg



KARTÄUSERSTRASSE 59
D-7800 FREIBURG/BRST.
TELEFON 0761-382038

Bei uns werben bringt
GEWINN
Sprechen Sie mit uns.
Heim Verlag ☎ 06151/56057
BUF

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 07751/3094

7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Hafenbad 35 · 7900 Ulm/Donau
Telefon (0731) 62699

EDV-Systeme
Software-erstellung
Schulung

Systemhaus:
Frauenstraße 28
7900 Ulm/Donau
Tel. (0731) 28076
Telex 712973 csulm-d

COMPUTER STUDIO GmbH

7918 Illertissen

biotech gmbh
technische Informationsysteme
Computerladen
Marktplatz 13
7918 Illertissen
07303/5045

7980 Ravensburg

GRAHLE
Expert Grahle Computer
Eisenbahnstr. 33
7980 Ravensburg
Tel.: 0751/15955
Vertragshändler für ATARI, Schneider und Star

8000 München

star · NEC



HDS-COMPUTER-VERTRIEBS GMBH
Am Klostergarten 1 · 8000 München 60
Tel. (089) 837063-64
ATARI · Commodore

Ludwig
COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STRASSE 62L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

schulz computer
Schillerstraße 22
8000 München 2
Telefon (089) 597339
Beratung · Verkauf · Kundendienst

8032 Gräfelfing



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464,851043

8100 Garmisch-Partenk.

Uwe Langheinrich Elektronik Center
Hindenburgerstr. 45
8100 Garmisch-Partenkirchen
Tel. 08821-71555
Bitte Gratisliste anfordern

8150 Holzkirchen



ATARI

Besuchen Sie uns!

Fordern Sie unseren Software-Katalog (520ST) an!

MÜNZENLOHER GMBH
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (0 80 24) 18 14

8170 Bad Tölz

Uwe Langheinrich Elektronik Center
Wachterstr. 3
8170 Bad Tölz
Tel. 0 80 41 - 4 15 65
Bitte Gratisliste anfordern

8330 Eggenfelden



Computer-Centrum
R. Lanfermann

Schellenbruckstraße 6
8330 Eggenfelden
Telefon 0 87 21 / 65 73

Altöttinger Straße 2
8265 Neuötting
Telefon 0 86 71 / 7 16 10

8400 Regensburg

Zimmermann elektroland

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85

8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

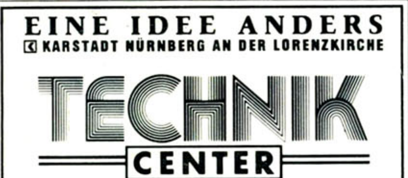
8423 Abensberg

COMPUTERVERSAND WITTICH
Tulpenstr. 16 · 8423 Abensberg
☎ 0 94 43 / 453



8500 Nürnberg

EINE IDEE ANDERS
KARSTADT NÜRNBERG AN DER LORENZKIRCHE



1. KLASSE EINKAUFEN IM WELTSTADTHAUS

hib
HIB Computer GmbH
Äußere Bayreuther Str. 57a – 59
8500 Nürnberg 10
Tel.: (09 11) 56 29 26 · Telex: 17 – 91 18 253
Teletex: 2627 – 91 18 253 · Telefax: (09 11) 51 30 40
Systemfachhändler für anspruchsvolle Computertechnik
SONY ATARI brother Microsoft TOSHIBA ZENITH data systems EPSON

8520 Erlangen



Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76

Zimmermann elektroland

8520 Erlangen
Nürnberger Straße 88
Tel. (0 91 31) 3 45 68

8500 Nürnberg
Hauptmarkt 17
Tel. (09 11) 2 07 98

8600 Bamberg



Bamberg · Tel. 0951 / 2 78 08 - 09

8700 Würzburg



Hardware · Software
Service · Schulung

computer center
am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 65 21 54

8900 Augsburg



Unser Plus: Beratung u. Service
Schwalbenstr. 1 · 8900 Augsburg-Pfersee
Telefon (08 21) 52 85 33 oder 52 80 87
Computer Vertriebs- und Software GmbH

ÖSTERREICH

A-1040 Wien

Ihr ST-Fachhändler in Wien
Computer-Studio
Wehsner Gesellschaft m.b.H
A-1040 Wien · Paniglgasse 18-20
Tel. (02 22) 5 05 78 08, 5 05 88 93

A-8010 Graz



ATARI
SUCO-COMPUTER
8010 Graz, Grazbachgasse 47
Tel. (0 31 6) 76 4 61

SCHWEIZ



Computer Trend

Ihr Computer Spezialist

5000 Aarau, Bahnhofstrasse 86,
Tel. 064/22 78 40
4102 Basel-Binningen, Kronenplatz,
Tel. 061/47 88 64
5430 Wettingen, Zentralstrasse 93,
Tel. 056/27 16 60
8400 Winterthur, St. Gallerstrasse 41,
Tel. 052/27 96 96
8021 Zürich, Langstrasse 31,
Tel. 01/241 73 73

Grösste Auswahl an Peripherie, Software, Literatur und Zubehör.

CH-1205 Geneve

PIMENT ROUGE
INFORMATIQUE S.A.
8, RUE DES MARAICHERS
1205 GENEVE TEL. 022/28 56 24

CH-1700 Fribourg

FRIDAT SA INFORMATIQUE
ehem. Softy Hard's Computershop

VOTRE SPECIALISTE
Rte des Grives 4
1700 Granges-Paccot/Fribourg
Tel. 0041 (0)37 26 66 28
Fax. 0041 (0)37 26 61 06

CH-2503 Biel



URWA ELECTRONIC
Computer Hard- und Software

Ihr ATARI ST Spezialist
in der Schweiz.
☎ 032/41 35 35

Bözingenstraße 133, 2504 Biel

CH-4313 Möhlin

BCR Computerdienst
Bahnhofstrasse 63
CH-4313 Möhlin

Computersysteme
EDV-Beratung
Installationen
CAD Anlagen
Datenpflege + Service

ATARI
NEC
star

Tel. 061 88 30 32 FAX 061 88 30 03

CH-4625 Oberbuchsiten

STECTRONIC M. Steck
Electronic-Computer-Shop

Hauptstr. 104/137
CH-4625 OBERBUCHSITEN
Tel. 062/63 17 27 + 63 10 27

CH-8001 Zürich

ADAG LASERLADEN

SEILERGRABEN 41. 8001 ZÜRICH
MO-FR:12-18.30 TEL.01/251 49 34

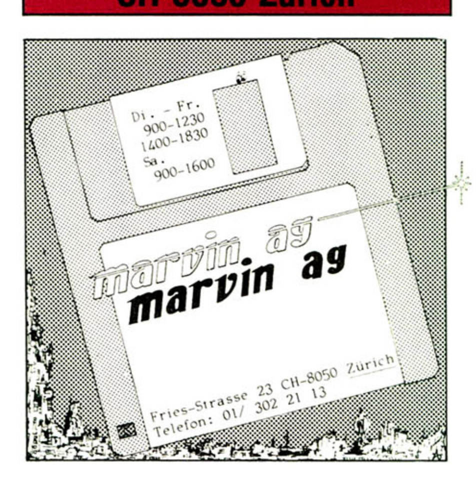
DTP... LASERPRINTS ... ET CETERA!

CH-8006 Zürich

ADAG
COMPUTER-SHOP
UNIVERSITÄTSSTR. 25
8006 ZÜRICH TEL. 01/252 18 68

Computer-Center P. Fisch
Stampfenbachplatz 4
8006 ZÜRICH
✆ 01/363 67 67

CH-8050 Zürich



Bei uns werben bringt

GEWINN



Sprechen Sie mit uns.
Heim Verlag 0 61 51 / 56057

BUF

CH-8052 Zürich

ACS COMPUTER
vortex FESTPLATTE HD
Star-Writer

Grünheidenstr. 28/Eisenbahnbrücke
8052 Zürich-Seebach
Tel: 01-302 2600 FAX: 01-301 4440

CH-8200 Schaffhausen

ZIMELEC
CAR-AUDIO
COMPUTER + ELECTRONIC
Bachstraße 28 · 8200 Schaffhausen
Tel.: 053/55224

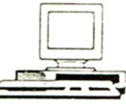
Montag – Freitag 9.00 – 12.00, 13.30 – 18.30
Samstag 9.00 – 16.00

CH-9000 St. Gallen

ADAG
COMPUTER-SHOP
TORSTR. 25
9001 ST.GALLEN
TEL. 071/25 43 42

CH-9400 Rorschach





Computer & Software
Kirchstrasse 38
CH-9400 Rorschach
Tel. 071/41 18 85

SIEMENS TOSHIBA ATARI PHILIPS brother EPSON star

PAUS-electronic
Hardware Software Systementwicklung

LUXEMBURG

Ihr Spezialist + Service für



bürodatik

Kleinanzeigen

BIETE HARDWARE

★★★★★★★★★★★★★★
★ – Kleine Preise – ★
★ – kleine Anzeigen – ☆
★ also: 1040 STF/124: 1.427,– ★
★ ST 2/4 m.124: 2.487/3.397,– ★
★ Megafile 30/60: 1.247/1.747,– ★
★ 200 dpi Scanner 997,– weit. ★
★ auf Anfrage. Fa CompuDruck ★
★ 02323/46362 Btx: 0232346329 ★
★★★★★★★★★★★★★★

NEC P6 und Brother HRIO-Typenrad gegen Höchstgebot 077617677

+ + + Hard- uznd Software + + +
z. B. HDplus 20 für DM 1145,–
Floppy 3.5-Zoll DM 299,–
Info: hl-Data, Lothar Ullmann
Birkenstr. 18, 7332 Eislingen
Tel. 07161/83406

Atari ST4, s/w Mönit., Laserdrukker, Vortex 60MB Platte, Signum2 Stad, PC-Ditto, neuwertig umständehalber, VHB 9000,–, 0622146430

BlitterROM-TOS 99 DM
02630/7525

Virus-Falle Hardware 49,95 DM
Eprom-Brenner für ST 249,00 DM
Roßmöller, Meckenheim, 02225206

Festplatten für Atari
Atari: 260ST – Mega-4
Anschlußfertig im Gehäuse mit Treibersoftware (nach CT)
z. B. 20 MB netto für DM 900,–
größere Kapazitäten verfügbar
RCE-GmbH, Witschelstraße 71,
8500 Nürnberg, ☎ 0911/326249 1-6

ST im IBM-Gehäuse 06144/1716

68020/25 MHz u. 68881/20 MHz sowie größere Anzahl RAM's abzugeben. Alles neu, Preis VB.
Tel.: 08031/88866

★★Neue Tastatur für Atari ST
★★ Die preiswerte & professionalle Lösung auch zum Selbstbau.
Info gegen frank. Rückumschlag von J. Hecker · An den Eichen 6
2000 Norderstedt

BIETE SOFTWARE

ST liest/beschreibt HP-format. Disketten. Info: Dipl.-Ing.
M. Poech 0711/297149 (18–22 h).

$Turbo-PD-Versand M. Gruber$
$ – Versand in 48 Stunden $
$ – Alle ST-Disks auf Lager $
$ – Virenfrei/Verify/updated $
$ – Nur Markendisketten $
$ – 1DD-Disk 5,10 DM $
$ – 2DD-Disk 8,10 DM $
$ (frei kombinierbar!) $
$ – laufend Sonderangebote $
$ – Versand/Porto 4,00 DM $
$ – PD-LISTE gegen 1,00 DM $
$ Am Weingert 27, D-8411 $
$ Pettendorf Tel. 09409/2271 $

Beckertext 2.0, or., 180 DM. **1st Prop** or., 50 DM. 069/880257

2 PDS von ST-Comp. auf 2-s. Markendisk. DM 5,- + NN DM 5,-
soft-STation T. 07195/53707

Lohn-Einkommensteuer, Miet-Lastenzuschuß, Rentenber./Beamtenversorgung.
H-I-SOFTWARE
Niederfelderstraße 44
8072 Manching · Tel. 08459-1669

Public-Domain-Blitzversand!
Riesenauswahl! Preise:
ab DM 4,– einseitige Disketten
ab DM 6,– doppelseitige
Inclusive Diskette!!!
Auch alle ST-Disks! Auf Doppeldisks beliebig kombinierbar!
Gratisliste anfordern bei:
A. Gauger Software
Buhlstraße 16a, 7505 Ettlingen
0 72 43 / 3 18 28
Bitte Computertyp angeben!!

Assembler (Profimat), Hard Disc Accelerator, The Lurking Horror Tel. (07131) 72639 16–20 Uhr

Vervollständigen
Sie schnell & unkompliziert
Ihre Atari-PD-Sammlung
5.00 pro Disk + NN
Klemens Köhler · Rhönstr. 6
6987 Kulsheim · 09345/1427

★ Professionelles Disketten- ★ Catalog-System. Automatische Disketten-Analyse. Anzeige und Druck nach unterschiedlichen Sortierbegriffen. Liste und Etiketten frei konfigurierbar. Eigene GEM-Oberfläche mit beeindruckender Graphik. Ausführliches deutsches Handbuch. DM 49,– + Versandkosten von DM 5,– bei Scheck bzw. DM 10,– bei Nachnahme.
kb-soft K. Beyer 2000 Wedel
Rissenerstr. 70

Ultra-Card V1.0 Die neue Dimension der Datenverarbeitung! 62 500 Datenfelder mit max. 1900 MB (!) je Datenfile! **Info anfordern!**
Language-Profi V1.0 Das Übersetzungs-Programm! Übersetzt jeden Text vom Deutschen ins Engl. und umgekehrt! **DM 79**
Bestellungen: Mit V-Scheck an:
M.K-S / Gronauer 19 / 48 Bielefeld

Easy Rider - Reassembler - V1.1
EASY RIDER erstellt von jedem Programm einen editier- und assemblierfähigen Quelltext! DM 159,–
★★★ Und Neu!! ★★★
Easy Rider - Der Assembler - V1.0
Macros, Include, Linker, Objektmodus und unglaublich schnell!!
DM 129,– A. Borchard 0541/87024
Wiesenbachstr. 2a · 45 Osnabrück

PD-Grafik, DTP, Art-Lib für Degas +Stad. Katalog: Frey, Rheinstr. 12A 6538 Münster-Sarmsheim

Lohn- und Einkommensteuer '88
+ Tabellen + Was wäre wenn
Mandantenfähig, Berlin **50 DM**
Dipl. F. J. Höfer, Grunewald 2a
5272 Wipperfürth 02192/3368

Softscroll-Laufschrift als GFA-Procedur, Pos. frei bestimmbar, bis 32 Kb Text darstellbar!
07147/4747 .LST-FILE 39,– DM

■■ Brandneue PD-Software ■■
■ selbstverständlich Virenfrei ■
■ **GRATIS** katalog beim ■
■ ST profi-partner ■
Mönkhofer Weg 126, 24 Lübeck 1
■■ Telefon: 0451-505367 ■■

CAD Projekt, GfA-Basic, Compiler Profi-Text-Mod. ST Lit. Neupreis 650,– jetzt 300,– 0871/34948

ORIGINALPROGRAMME
Riesenauswahl an neuesten Programmen zu den günstigen – Gauger Software Preisen –
... die Gelegenheit!!
Gratisliste anfordern bei:
A. Gauger Software, Buhlstr. 16a
7505 Ettlingen, 07243/31828

GROSSE GRAFIKSAMMLUNG!
ca. 2000 IMG-Grafiken (z. B. für Wordplus) DM 25,– 06302-3338

Programme f. ST zu verk. z.B.
GTI-Datenbank 150,–
Tel. 07634-2195

■ Verk. ges. PD-Samml. gg. ■
■ Höchstgeb. INFO > ■
■ G. Grasberger, C-Orff-Str. 9 ■
■■ 8221 Traunwalchen ■■

!!!!!!!!!!!!! **PD-Software** !!!!!!!!!!!!!
! Programme kompromißlos zu-
! sammenstellbar nach Kilobyte.
! Katalog bei: O. Schwede,
! Röntgenweg 9/1, 7050 Waiblingen

★★★★ PD-SERVICE ★★★★
★ Einzeldiskette SS **DM 4,00** ★
★ Doppeldiskette DS **DM 6,00** ★
★ beliebig kombinierbar) ★
★ Porto u. Verpack. DM 3,00 ★
★ V.-Scheck o. NN (+DM 3,50) ★
★ N. Twardoch, Gröchteweg 22 ★
★ 4900 Bad Salzuflen 1 ★

★★PD für ATARI-ST, IBM★★
Atari(450), IBM(2300), **3–6 DM**
Info frei, Kat. PC-10DM, ST-5DM
Johrend, Neusalzer Str. 9, 85 Nbg

LERNPROGRAMME
für den ATARI ST
1. Kommerzielle ST-Lernprogramme
2. PD-Lernprogramme für den ST
Katalog bei: **LTA-LERNSOFT**
Friedenstr. 7 · 8032 Lochham
089/8545531

■**Fußball-Bundesliga** für alle ST.
■ Alle Ergebnisse, Tabellen und
■ Statistiken seit 1964! Leicht zu
■ aktualisieren. Für jeden Drucker.
■ Mausbedienung. Gratisinfo
■ anfordern oder gleich bestellen.
■ 40,- DM Bar, Scheck + NN.
■ Andreas Smoor, Tannenstr. 50
■ 4460 Nordhorn

★★ Public Domain Software ★★
IBM u. komp. Disk nur 5,– DM
Atari ST Disk ab 3,– DM
Riesenauswahl ★★★ Superpreise
Katalog-Disk gegen Rückporto
Graf & Schick · Hauptstraße 32 a
8542 Roth · Tel. 0 91 71 / 50 58 - 59

Wärmebedarf DIN4701 + K-Zahl
★ Heizflächenauslegung ★ Rohrnetz ★ Demodisk 2-seitig für DM 10 Vorkasse von J. Binder, Eichendorffstr. 15 · 5030 Hürth

■ ■ **Public Domain** ■ ■
■ ■ **Software** ■ ■
■■ für alle ■■
■ ■ Atari ST-Modelle ■ ■
■ ■ ■ ■
Sie suchen aktuellste PD-Soft?
Sie interessieren sich für
digit. Diashows, Animationen?
Dann fordern Sie unseren neuen „virenfreien" Gratiskatalog an
Klaus Kohler
Don-Carlos-Str. 33 B
7 Stuttgart 80

IDEE
Individuelle Computer-Lösungen
HAUSVERWALTUNG: 398,- DM
Tel.: 040/ 38 13 64

PLOT_ST für Atari-Fortran!
Einfache Ausgabe grafischer Darstellungen auf Bildschirm und Drucker für 95,– DM. Kompatibel mit Großrechner Standard (Plot-79) Handbuch DM 15,– (wird verrechnet); Info gratis. PLOT-ST sowie weitere FORTRAN-Tools bei:
M. Gamer, Friedrichsring 26,
6050 Offenbach

★ST-PD 3,5" + 5,25" ★ab 2 DM
★ z. B. alle PD aus ST-Computer u.a. ★ Info: T. Helfers
Portslogerstr. 30, 2905 Edewecht
T: 04405/6809

■ Mühleprogramm »Ramses« ■
Spielstark, Mattsuche, Editor...
DM 29,90 + NK Tel. 0831/16993
US-Soft, Pf 2304, 8960 Kempten

Orig. ST-Base III, Tel. 061441716

PD-Software ab 2,– DM/Diskette von PD 1 bis heute. Auch viele „eigene" PD's lieferbar.
Tel. 02721/2432 von 12 bis 22 Uhr

■■■Strucadi ■■■ Strucadi ■■■
Nie mehr Einrücken! Aditalk-Programme werden übersichtlich!!!
Preis: 30,– Reichel, 7500 K'he-41
Durlacher Str. 23 · 0721/481438

■■ Talk-XREF ■ Talk-XREF ■■
Crossreference für ADITALK.
Variablenlisten n. Importiert. 40,–;
Demo 15,–; Reichel, 7500 K'he 41
Durlacher Str. 23 · 0721/481438

Reassembler: Erstellt Quell- + Labeldatei, Labeldatei auch f. Eingabe, **40 DM** bei Verr. Scheck, sonst + NN. Bernd Hofmann
Oberfeldstr. 9, 6633 Wadgassen

SUCHE HARDWARE

Kaufe defekte ATARI-ST und Zubehör. Angebot an 07634-2195

Kaufe STs (a. def.) ☎: 0431569216

SUCHE SOFTWARE

Cashflow Kassenbuch 05138/4540

Suche Anleitung für Hausverwaltung „Date Becker" mit oder ohne Diskette

Su. Prg. z. Wohnungsverw. 10 Einh.
Tel. 04521/9553

TAUSCH

■PD-Tausch gesucht 0571-75377■

KONTAKTE

Philos. m. fasz. Perpekt. d. **KI** sucht Program.-Erf. m. Masch.-Nähe zu 6800/Atari. Möchte meine Gedanken zur **Neurologie** in eine Bsp.-Disk f. Buch umsetzen (ähnl. DNS-AMS-Code!).

Suche Partner/Käufer zur Vermarktung von Exist. 66K-Prg. Das RSC-Files in Mod2 wandelt.

Schreibe Software, die speziell auf Ihre Bedürfnisse zugeschnitten ist.
Tel.: 02175/71364

VERSCHIEDENES

★ Atari ST im PC Gehäuse ★
Individueller Umbau
Info gegen 1,30 DM in Briefmarken bei: T. Allendorf,
Elbestr. 2, 6096 Raunheim/M.
★ Atari ST im PC Gehäuse ★

Verkaufe ALLE ST-COMPUTER
DM 120,– Tel.: 089/5804646

Sammelbestellung des Atari TT. Durch Großeinkauf billiger Lieferung mit/ohne 1280×960 Monitors. TT oder TT Unix
Atari TT · Postfach 1253
8425 Neustadt/Donau

Suche & biete PDs; Pürkner, Kurt-Eisner-Str. 27, 8000 München 83



Programmier Praxis

Auch wenn es noch nicht ganz soweit ist, möchten wir Ihnen doch für das neue Jahr weiterhin interessante Beiträge zur Programmierpraxis versprechen. Diesmal haben wir Beiträge für fast alle gängigen Programmiersprachen.

Einen kleinen Beitrag in Numerischer Mathematik haben wir für die Berechnung von sogenannten Splines. Dabei werden von einem Polygonzug die Eckpunkte genommen und diese dann interpoliert. Dadurch ist eine sichtbare Glättung der Kurve zu erreichen.

Beim Anzeigen verschiedener Farbbilder passiert es oft, daß sie über verschiedene Farbpaletten verfügen. Normalerweise kann man diese Paletten einfach umschalten. Will man allerdings die Bilder miteinander mischen, tritt ein Problem auf. Unser Beitrag "Mit fremden Farben gemalt schafft Abhilfe.

Mit dem Universal-ASCII-Datei-Formatierer lassen sich frei wählbare Worte in einem beliebigen Text groß schreiben. Zusätzlich kann man den Text auch mit Zeilennummern versehen. Praktisch um z.B. ein Listing übersichtlicher zu gestalten.

Im letzten Beitrag kommen dann die BASIC-Programmierer unter unseren Lesern zum Zug. Hier geht es um das Installieren von Zeichensätzen des Grafikprogrammes STAD als Systemzeichensatz. Benutzt wird dazu der GFA BASIC-Befehl DEFTEXT.

INHALT



RUND muß es sein

Dietmar Rabich

Polygonzüge haben manchmal einen Nachteil - es sind Polygonzüge! Das hier vorgestellte Programm nimmt dem Polygonzug seine Ecken, es interpoliert zwischen den einzelnen Punkten.

Eine Kurve ohne die von Polygonzügen typischen Ecken ist nicht ohne eine zusätzliche Berechnung möglich. Interpolation bietet sich geradezu an, um eine Kurve sichtbar zu glätten. Vielleicht werden derartige Routinen auch mal in einigen Graphikprogrammen aufgenommen!

Aber erst eine kurze Einführung in die mathematischen Notwendigkeiten. Interpolation wird benötigt, wenn man statt einer Funktionsvorschrift nur einzelne Werte kennt, beispielsweise bei einer Reihe von Meßwerten. Mit Hilfe mathematischer Formeln werden Zwischenwerte berechnet, so, wie sie tatsächlich sein könnten. Zumindest sollte eine Interpolation so gute Werte liefern, daß sie von tatsächlichen nur minimal abweichen.

Bei unserem Programm werden ebenfalls Werte angenähert. Vorgegeben sind eine Reihe (mindestens vier) von Wertepaaren für Bildschirmkoordinaten. Durch diese Punkte wird eine Kurve gelegt. Das Beispielprogramm liefert schon eine recht anschauliche Kurve. Die Marker sollen zeigen, welche Punkte vorgegeben waren. Es sind wirklich nur zwölf!

Gerade war schon von Funktionswerten die Rede. Sie sind hier - getrennt nach x- und y-Koordinaten - die Bildschirmkoordinaten, also einerseits nacheinander die x-Koordinaten der Punkte und andererseits die y-Koordinaten. Neben diesen Werten sind noch Ableitungen nötig. Die erste Ableitung einer Funktion sagt etwas über ihr Wachstumsverhalten aus, die zweite etwas über ihr Krümmungsverhalten - und genau das interessiert uns! Wir brauchen nunmehr neben den Bildschirmkoordinaten auch das Krümmungsverhalten der Kurve in den jeweiligen Punkten.

Das ist nicht so schlimm, denn die Aufgabe der Berechnung übernimmt das Programm. Allerdings müssen zwei Werte vorgegeben werden, das Programm kann das Krümmungsverhalten am Anfangs- und Endpunkt der Kurve nicht berechnen. Aber zwei Werte sind ja leicht anzugeben, notfalls nimmt man immer 0, was bedeutet, daß die Kurve in den beiden angesprochenen Punkten nicht gekrümmt ist.

Bevor die Kurve ausgegeben werden kann, ist die Näherung zu berechnen. Diese Aufgabe übernimmt die Routine *MakeSplines*. Die Ausgabe übernimmt schließlich *DrawSplines*. Mehr ist bei der Bedienung nicht zu beachten.

DrawSplines ist den VDI-Routinen von Megamax Modula 2 angepaßt. Daher wird die Verwendung einfacher. Wer die Graphikfunktionen des VDI nicht ausnutzen möchte, kann die Ausgabe ja auch mit LineA-Routinen abwandeln.

DrawSplines hat, wie schon angesprochen, die Aufgabe, die Kurve auszugeben. Die bereits berechneten Werten werden in die Spline-Interpolierende (das Programm benutzt Spline-Interpolation), ein kubisches Polynom, eingesetzt und ausgewertet. Die Auswertung ist lediglich die Ausgabe der Punkte, d.h. Linien zwischen den einzelnen Punkten. Wieviele interpolierte Punkte berechnet werden sollen, ist auch in der Routine *DrawSplines* anzugeben. Da mit Linien gearbeitet wird, brauchen es nicht sehr viele zu sein. Je mehr Punkte berechnet werden sollen, desto länger dauert die Ausgabe auf dem Monitor.

Die beiden Routinen *Square* und *Round* werden nur intern benötigt. Da sie aber auch von allgemeinem Interesse sein könnten, sind sie auch im Definitionsmodul eingetragen.

Die Hauptaufgabe hat *Make Splines*. Zuerst werden Parameter berechnet. Hat man beispielsweise eine Funktion $f(x)$ gegeben, so existieren neben den Bildwerten $f(x)$ auch Urbildwerte x. Die Bildschirmkoordinaten werden nur als Bildwerte angesehen, was fehlt, sind Urbildwerte.

MakeParameter liefert die Urbildwerte, einfach den Abstand der Punkte nacheinander. Danach wird in der Routine *Make-System* das Gleichungssystem aufgebaut, welches als Lösung das Krümmungsverhalten der Kurve an den einzelnen Punkten liefert, oder, um es noch etwas genauer auszudrücken, es wird die zweite Ableitung an den einzelnen Stellen berechnet. Glücklicherweise hat die Koeffizientenmatrix dieses linearen Gleichungssystems Tridiagonalgestalt, wodurch sich

der Rechenaufwand verringert.

Bevor man die Koeffizienten des Interpolationspolynoms in *SetSplines* errechnen kann, muß noch mit *SolveSystem* die Lösung des Gleichungssystem ermittelt werden.

Die Koeffizienten des Interpolationspolynoms sind *a,b,c* und *d*, wobei *a* bei dem Term mit Grad 3 steht. In der *DrawSplines*-Routine ist dies deutlich zu erkennen. Wer mehr zur Theorie der Spline-Interpolation oder zu den Graphikfunktionen des VDI wissen möchte, der findet in den beiden unten angeführten Büchern sicherlich wertvolle Hinweise.

Das Listing beinhaltet keine extremen Besonderheiten von Modula-2. Eine Umsetzung in andere Programmiersprachen dürfte somit nicht schwer sein.

Literatur:

[1] Numerische Mathematik, H.R. Schwarz, Teubner

[2] Atari ST Profibuch, H.-D. Jankowski/J.F. Reschke/D. Rabich, Sybex



```
 1:  (***********************************************)
 2:  (* Berechnung von Splines in der Ebene zur
        Glättungvon Polygonzügen oder zur          *)
 3:  (* Verbindung einzelner Punkte                *)
 4:  (* -------------------------------------------*)
 5:  (* Definitionsmodul, Version 1.0, 27. April 1988*)
 6:  (* -------------------------------------------*)
 7:  (* Autor: Dietmar Rabich, Dülmen               *)
 8:  (* -------------------------------------------*)
 9:  (* Entwickelt mit Megamax Modula 2 von Application
        Systems /// Heidelberg.    *)
10:  (***********************************************)
11:
12:  DEFINITION MODULE Splines;
13:
14:  FROM GrafBase IMPORT Point;
15:  FROM GEMEnv   IMPORT DeviceHandle;
16:
17:  CONST MaxPoint = 25; (* Maximalanzahl der Punkte*)
18:
19:  TYPE PointsArray     = ARRAY [0..MaxPoint-1] OF
                           Point; (* Punkte *)
20:       SplinesRecord   = RECORD (*Record für Splines*)
21:                          a,b,c,d,
                        (* Koeffizienten des Polynoms *)
22:                        XStart,XEnde : REAL
                        (* Definitionsbereich          *)
23:                        END;
24:       SplinesArray    = ARRAY [0..MaxPoint-1] OF
                           SplinesRecord;
25:
26:  (* Quadrat *)
27:  PROCEDURE Square (x : REAL) : REAL;
28:
29:  (* Rundung *)
30:  PROCEDURE Round (x : REAL) : LONGINT;
31:
32:  (* berechnet Interpolationspolynom *)
33:  PROCEDURE MakeSplines (pts              :
PointsArray;
34:                         NoPts            : CARDINAL;
35:                         VAR splA,splB
:SplinesArray;
36:                         ysa,ysb          : REAL);
37:
38:  (* wertet Polynom aus, läuft nur unter GEM
        (mit VDI) *)
39:  PROCEDURE DrawSplines (dev        : DeviceHandle;
40:                         splX,splY  : SplinesArray;
41:                         No         : CARDINAL);
42:
43:  END Splines.
44:
45:  (* Modulende *)
```

Listing 1: Das Definitionsmodul

```
 1: (***********************************************)
 2: (* Berechnung von Splines in der Ebene
       zur Glättung von Polygonzügen oder zur       *)
 3: (* Verbindung einzelner Punkte                  *)
 4: (* ---------------------------------------------*)
 5: (* Implementationsmodul,Version 1.1, 24. Mai 1988*)
 6: (* ---------------------------------------------*)
 7: (* Autor: Dietmar Rabich, Dülmen                *)
 8: (* ---------------------------------------------*)
 9: (* Entwickelt mit Megamax Modula 2 von Application
       Systems /// Heidelberg.                       *)
10:  (***********************************************)
11:
12:  IMPLEMENTATION MODULE Splines;
13:
14:  (* Importe *)
15:  FROM GrafBase    IMPORT Pnt,Point;
16:  FROM GEMEnv      IMPORT DeviceHandle;
17:  FROM MathLib0    IMPORT sqrt,power,real,entier;
18:  FROM VDIOutputs  IMPORT Line;
19:
20:  (* Typen *)
21:  TYPE ParameterArray = ARRAY [0..MaxPoint-1]
                          OF REAL;  (* Parameter *)
22:       ErgArray       = ParameterArray;
                          (* Ergebnis (y'') *)
23:
24:  (* Quadrat *)
25:  PROCEDURE Square (x : REAL) : REAL;
26:
27:   BEGIN
28:    RETURN power(ABS(x),2.0)
29:   END Square;
30:
31:  (* Rundung *)
32:  PROCEDURE Round (x : REAL) : LONGINT;
33:
34:   BEGIN
35:    RETURN entier(x+0.5)
36:   END Round;
37:
38:  (* Addition von y zu x *)
39:  PROCEDURE IncReal (VAR x : REAL; y : REAL);
40:
41:   BEGIN
42:    x:=x+y
43:   END IncReal;
44:
45:  (* Routine, die Splines (Koeffizienten eines
        kubischen Interpolations- *)
46:  (* polynoms) berechnet. *)
47:  PROCEDURE MakeSplines (pts          : PointsArray;
48:                         NoPts        : CARDINAL;
49:                         VAR splA,splB:SplinesArray;
50:                         ysa,ysb      : REAL);
51:
52:   TYPE GlSysArray = ErgArray;
53:        TriGlSys   = RECORD
                   (* u/o : untere/obere Diagonale *)
54:                  o,u,m,l : GlSysArray
                   (* m   : Hauptdiagonale  *)
55:                  END;
56:                (* l   : Inhomogenität     *)
57:   VAR Param           : ParameterArray;
58:       SystemA,SystemB : TriGlSys;
59:       ergA,ergB       : ErgArray;
60:
61:   (* berechnet Parameter, damit die Koordinaten als
        normale Stützwerte *)
62:   (* genutzt werden können                       *)
63:   PROCEDURE MakeParameters (p     : PointsArray;
64:                             VAR t : ParameterArray;
65:                             No    : CARDINAL);
66:
67:    VAR i : CARDINAL;
68:
69:    BEGIN
70:     t[0]:=0.0;        (* bei 0 geht's los *)
71:     FOR i:=1 TO No-1 DO
72:      t[i]:=t[i-1]+      (* alter Wert +     *)
73:            sqrt(Square(real(p[i].x-p[i-1].x))+
```

```
              (* Abstand der Punkte *)
 74:             Square(real(p[i].y-p[i-1].y)))
 75:     END
 76:    END MakeParameters;
 77:
 78:   (* berechnet lineares Gleichungssystem mit
         tridiagonaler *)
 79:   (* Koeefizientenmatrix      *)
 80:   PROCEDURE MakeSystem (VAR system1,system2 :
                                  TriGlSys;
 81:                        p   : PointsArray;
 82:                        t   : ParameterArray;
 83:                        No  : CARDINAL);
 84:
 85:    VAR h : ParameterArray;
 86:        i : CARDINAL;
 87:
 88:    BEGIN
 89:     FOR i:=0 TO No-2 DO      (* Schrittweiten *)
 90:      h[i]:=t[i+1]-t[i]
 91:     END;
 92:     FOR i:=0 TO No-4 DO      (* Haupt- und
                                    Nebendiagonale *)
 93:      system1.m[i]:=2.0*(h[i]+h[i+1]);
 94:      system1.o[i]:=h[i+1];
 95:      system1.u[i]:=system1.o[i]
 96:     END;
 97:     system1.m[No-3]:=2.0*(h[No-3]+h[No-2]);
 98:     system2         :=system1;    (* 2. System *)
 99:     FOR i:=0 TO No-3 DO          (* Inhomogenität *)
100:      system1.l[i]:=6.0*(real(p[i+1].x-p[i].x)/h[i]
101:                    -real(p[i+2].x-p[i+1].x)h[i+1]);
102:      system2.l[i]:=6.0*(real(p[i+1].y-p[i].y)/h[i]
103:                    -real(p[i+2].y-p[i+1].y)/h[i+1])
104:     END;
105:     IncReal(system1.l[0],    h[0]    *ysa);
         (* Randwerte dazu *)
106:     IncReal(system1.l[No-3],h[No-2]*ysb);
107:     IncReal(system2.l[0]    ,h[0]    *ysa);
108:     IncReal(system2.l[No-3],h[No-2]*ysb)
109:    END MakeSystem;
110:
111: (* berechnet Lösung des Gleichungssystemes          *)
112: (* stark vereinfacht, da Matrix Tridiagonalmatrix*)
113:   PROCEDURE SolveSystem (system   : TriGlSys;
114:                          VAR erg  : ErgArray;
115:                          No       : CARDINAL);
116:
117:    VAR i,Back : CARDINAL;
118:        m,l,x  : ErgArray;
119:
120:    BEGIN
121:     m[0]:=system.m[0];          (* LR - Zerlegung *)
122:     FOR i:=0 TO No-4 DO
123:      l[i]  :=system.u[i]/m[i];
          (* L - Matrix, Nebendiagonale *)
124:      m[i+1]:=system.m[i+1]-l[i]/system.o[i]
          (* R - Matrix, Hauptdiagonale *)
125:     END;
126:     x[0]:=system.l[0];      (* einsetzen            *)
127:     FOR i:=1 TO No-3 DO     (* berechnet x aus      *)
128:      x[i]:=system.l[i]-l[i-1]*x[i-1]
          (* L x -system.l = 0         *)
129:     END;
130:     erg[No-3]:=-x[No-3]/m[No-3]; (* Lösung          *)
131:     FOR i:=0 TO No-4 DO      (* berechnet erg aus *)
132:      Back:=No-4-i;            (* R erg+ x = 0      *)
133:      erg[Back]:=-(x[Back]+system.o[Back]
                   *erg[Back+1])/m[Back]
134:     END
135:    END SolveSystem;
136:
137:   (* berechnet die Koeffizienten des
         Interpolationspolynoms *)
138:   PROCEDURE SetSplines (VAR spl1,spl2    :
                                  SplinesArray;
139:                        erg1,erg2 : ErgArray;
140:                        par       : ParameterArray;
141:                        p         : PointsArray;
142:                        No        : CARDINAL);
143:
144:    VAR y    : ErgArray;
145:        i    : CARDINAL;
146:
147:    BEGIN
148:     (* 1. für x-Werte *)
149:     FOR i:=0 TO No-3 DO         (* 2. Ableitung  *)
150:      y[i+1]:=erg1[i]
151:     END;
152:     y[0]    :=ysa;
153:     y[No-1]:=ysb;
154:     FOR i:=0 TO No-2 DO
155:      WITH spl1[i] DO
156:       a      :=(y[i+1]-y[i])/(6.0*(par[i+1]-
                    par[i]));(* Koeffizienten *)
157:       b      :=y[i]/2.0;
158:       c      :=real(p[i+1].x-p[i].x)/(par[i+1]-
                    par[i])
159:                 -(par[i+1]-par[i])*(y[i+1]+2.0*y[i])/
                    6.0;
160:       d      :=real(p[i].x);
161:       XStart:=par[i];        (* Anfang *)
162:       XEnde :=par[i+1]       (* Ende   *)
163:      END
164:     END;
165:     (* 2. für y-Werte *)
166:     FOR i:=0 TO No-3 DO     (* 2. Ableitung  *)
167:      y[i+1]:=erg2[i]
168:     END;
169:     FOR i:=0 TO No-2 DO
170:      WITH spl2[i] DO
171:       a      :=(y[i+1]-y[i])/(6.0*(par[i+1]-
                    par[i]));(* Koeffizienten *)
172:       b      :=y[i]/2.0;
173:       c      :=real(p[i+1].y-p[i].y)/(par[i+1]-
                    par[i])
174:                 -(par[i+1]-par[i])*(y[i+1]+2.0*y[i])/
                      6.0;
175:       d      :=real(p[i].y);
176:       XStart:=par[i];   (* Anfang *)
177:       XEnde :=par[i+1]  (* Ende   *)
178:      END
179:     END
180:    END SetSplines;
181:
182:  BEGIN
183:   MakeParameters(pts,Param,NoPts);
184:   MakeSystem(SystemA,SystemB,pts,Param,NoPts);
185:   SolveSystem(SystemA,ergA,NoPts);
186:   SolveSystem(SystemB,ergB,NoPts);
187:   SetSplines(splA,splB,ergA,ergB,Param,pts,NoPts);
188:  END MakeSplines;
189:
190:  (* wertet Interpolationspolynom aus und stellt
        Kurve graphisch dar *)
191:  PROCEDURE DrawSplines (dev        : DeviceHandle;
192:                         splX,splY : SplinesArray;
193:                         No         : CARDINAL);
194:
195:  VAR i        : CARDINAL;
196:      x,Abst   : REAL;
197:      X,Y      : LONGINT;
198:      PMem,P   : Point;
199:      set      : BOOLEAN;
200:
201:  BEGIN
202:   set:=FALSE;
203:   (* alle Polynome nacheinander *)
204:   FOR i:=0 TO No-2 DO
205:    (* alle Werte von XStart bis XEnde *)
206:    Abst:=(splX[i].XEnde-splX[i].XStart)/20.0;
        (* Feinheit der Unterteilung  *)
207:    (* je feiner die Unterteilung, desto genauer
          wird die Kurve dargestellt, *)
208:    (* aber auch desto langsamer wird die
          Darstellung *)
209:    x:=splX[i].XStart;
210:    WHILE x<=splX[i].XEnde DO
211:
212:     (* Interpolationspolynome *)
213:     X:=Round(splX[i].a*power((x-splX[i].XStart),
          3.0)+ (* für x *)
214:         splX[i].b*Square(x-splX[i].XStart)+
215:         splX[i].c*(x-splX[i].XStart)+
216:         splX[i].d);
217:     Y:=Round(splY[i].a*power((x-splY[i].XStart),
          3.0)+ (* für y *)
218:         splY[i].b*Square(x-splY[i].XStart)+
219:         splY[i].c*(x-splY[i].XStart)+
220:         splY[i].d);
221:
```

```
222:        P:=Pnt(SHORT(X),SHORT(Y)); (* neuer Punkt   *)
223:        IF ~set THEN
224:         PMem:=P;    (* erster Punkt als Startpunkt *)
225:         set :=TRUE
226:        ELSE
227:         Line(dev,PMem,P);  (* Linie zu P ziehen    *)
228:         PMem:=P
229:        END;
230:        IncReal(x,Abst)
231:       END
232:      END
233:    END DrawSplines;
234:
235:  END Splines.
236:
237:  (* Modulende *)
```

Listing 2: Das Implementationsmodul

```
 1:  (*************************************************)
 2:  (* Berechnung von Splines in der Ebene zur
        Glättung von Polygonzügen oder zur           *)
 3:  (* Verbindung einzelner Punkte                  *)
 4:  (* ---------------------------------------------*)
 5:  (* Demoprogramm, Version 1.0, 27. April 1988    *)
 6:  (* ---------------------------------------------*)
 7:  (* Autor: Dietmar Rabich, Dülmen                *)
 8:  (* ---------------------------------------------*)
 9:  (* Entwickelt mit Megamax Modula 2 von
        pplication Systems /// Heidelberg.           *)
10:  (*************************************************)
11:
12:  MODULE SplinesDemo;
13:
14:  (* Importe *)
15:  FROM GrafBase        IMPORT Point,Pnt;
16:  FROM AESEvents       IMPORT KeyboardEvent;
17:  FROM GEMEnv          IMPORT InitGem,ExitGem,RC,
                                 CurrGemHandle,
18:                              DeviceHandle,GemHandle;
19:  FROM GEMGlobals      IMPORT GemChar,MarkerType;
20:  FROM VDIControls     IMPORT ClearWorkstation;
21:  FROM VDIAttributes   IMPORT SetMarkerType;
22:  FROM VDIOutputs      IMPORT PolyMarker;
23:  FROM Splines         IMPORT MaxPoint,PointsArray,
24:                              SplinesRecord,
                                 SplinesArray,
25:                              MakeSplines,DrawSplines;
26:
27:  VAR p          : PointsArray;
         (* Koordinaten der Punkte     *)
28:      n,  (* Anzahl der Punkte          *)
29:      i          : CARDINAL;       (* Laufvariable *)
30:      splX,splY  : SplinesArray;   (* Splines       *)
31:      yA,yB      : REAL; (* zweite Ableitung an
                               den Enden             *)
32:      gemHandle  : GemHandle;    (* Handle für GEM *)
33:      Device     : DeviceHandle;    (* Workstation *)
34:      Taste      : GemChar;         (* Taste       *)
35:      success    : BOOLEAN;    (* für Installation *)
36:
37:  BEGIN
38:   InitGem(RC,Device,success);
39:   IF success THEN                     (* GEM OK? *)
40:    gemHandle:=CurrGemHandle();        (* Handle  *)
41:    yA:=0.0;          (* 2. Ableitung= 0          *)
42:    yB:=0.0; (*  <=> keine Krümmung an den Enden *)
43:    n :=12;           (* 12 Punkte                *)
44:    p[0].x:=200;      (* Koordinaten *)
45:    p[0].y:=100;      (*      .      *)
46:    p[1].x:=100;      (*      .      *)
47:    p[1].y:=100;      (*      .      *)
48:    p[2].x:=100;
49:    p[2].y:=200;
50:    p[3].x:=200;
51:    p[3].y:=200;
52:    p[4].x:=200;
53:    p[4].y:=300;
54:    p[5].x:=100;
55:    p[5].y:=300;
56:    p[6].x:=300;
57:    p[6].y:=100;
58:    p[7].x:=400;
59:    p[7].y:=100;
60:    p[8].x:=350;
61:    p[8].y:=100;
62:    p[9].x:=350;
63:    p[9].y:=200;
64:    p[10].x:=350;
65:    p[10].y:=250;
66:    p[11].x:=350;
67:    p[11].y:=300;
68:    ClearWorkstation(Device);(* Bildschirm löschen*)
69:    SetMarkerType(Device,crossMark);(* Kreuz
                                          als Marker *)
70:    PolyMarker(Device,p,n);
         (* Marker setzen (nicht unbedingt nötig)   *)
71:    MakeSplines(p,n,splX,splY,yA,yB);(* Splines
                                           berechnen *)
72:    DrawSplines(Device,splX,splY,n); (* Splines
                                           ausgeben  *)
73:    KeyboardEvent(Taste);      (* auf Taste warten *)
74:    ExitGem(gemHandle)        (* Auf Wiedersehen! *)
75:   END
76:  END SplinesDemo.
77:
78:  (* Demoende *)
```

Listing 3: Das Demoprogramm

ENDE

Mit fremden Farben gemalt

Hans-Detlev Siewert

Der ATARI ST verfügt in seiner niedrigsten Auflösung bekanntlich über 16 Farben, die aus 512 Variationen frei gewählt werden können. Dies geschieht über indirekte Farbengebung, d.h. der Wert eines auf dem Bildschirm dargestellten Pixels (zwischen 0 und 15) dient als Offset in einer Tabelle, in der die aktuelle, eingestellte Palette abgelegt wurde. Meistens haben daher verschiedene Bilder auch höchst individuelle Paletten.

Schaltet man auf ein anderes Bild um, ohne gleichzeitig die Palette zu ändern, so erscheint es stark verfremdet, da es in den falschen Farben dargestellt wird. Das mag für Künstler (Surrealisten) interessant sein, für den normalen Anwender ist es zumindesten ärgerlich.

Normalerweise ist es ja auch ohne Problem möglich, die Palette umzuschalten, Was aber, wenn man mehrere Bilder mischen möchte? Ich denke da an viele Anwendungen wie z.B. bei Überblendungen, Raytracing etc.. Abgesehen von IMAGIC - ein Programm, das ich mir leider bisher noch nicht leisten konnte - ist mir kein Programm bekannt, das dies unterstützt.

An dieser Stelle springt mein Programm helfend ein. Es lädt ein Bild, rechnet es um und speichert es dann mit den neuen Farben ab. Zur Auswahl stehen zwei verschiedene Paletten: die Standard-Farbenpalette des Desktops und eine regenbogenähnliche. Durch einfache Änderungen des Programmes können Sie auch eigene Paletten einsetzen oder das Programm so umschreiben, daß die zu erzeugenden Farben von Diskette eingelesen werden bzw. aus anderen Bildern extrahiert werden können. Das Programm kann Bilder in dem weitverbreiteten DEGASformat verarbeiten.

Die Programmiersprache, die ich verwendet habe, ist Megamax-C. Sie können es leicht in andere C-Dialekte umschreiben, wenn diese Inline-Assemblerbefehle unterstützen oder auf die LineA-Grafikroutinen zugreifen können.

Zur Bedienung des Programmes ist nicht viel zu sagen: eine Fileselektorbox fragt nach dem Namen des Bildes, das dann eingelesen wird. Eine Alertbox fragt ab, welche der beiden Paletten verwendet werden soll. (Leider steht in der niedrigsten Auflösung dafür sehr wenig Platz zur Verfügung.) Danach erfolgt die Berechnung, die ca. 80 Sekunden dauert. Als letztes kann wieder in einer Fileselektorbox der Filename eingegeben werden, unter dem das neue Bild abgespeichert werden soll.

Das Programm benötigt nicht sehr viel Arbeitsspeicher- nur zur Einrichtung von zwei Arbeitsbildschirmen wird welches reserviert.

Nun zum verwendeten Algorithmus. Er versucht, mit der zur Verfügung stehenden Palette die größtmögliche Ähnlichkeit zwischen dem ursprünglichen und dem erzeugten Bild herzustellen. Dabei wählt er nicht nur für jedes Pixel die Farbe, die der ursprünglichen am ähnlichsten ist, sondern ist auch noch in der Lage, nicht vorhandene Farben durch additives Mischen der Farben von benachbarten Pixeln zu erzeugen. Wenn etwa Grün benötigt wird, dieses aber nicht in der Zielpalette vorhanden ist, werden ein blaues und ein gelbes Pixel nebeneinandergesetzt. Dadurch wird das umgesetzte Bild jedoch auch unschärfer, was leider nicht zu vermeiden ist. Auf manchen Monitoren - besonders Fernsehern - die gar nicht in der Lage sind, einzelne Pixel zu trennen, da sie schon von Natur aus unscharf sind, fällt dies nicht sehr stark ins Gewicht, aber solch ein Monitor wird dann auch kaum von ernsthaften Grafikfreaks benutzt.

Die Idee zu diesem Algorithmus lieferte mir ein Algorithmus von Floyd/ Steinberg, der kürzlich in der c't beschrieben wurde, jedoch nur zur Darstellung von Grauwerten auf reinen Monochrombildschirmen dient.

Der Algorithmus bearbeitet das Bild Zeile für Zeile, von oben nach unten. Ein Array enthält die Werte der Pixel im ursprünglichen Bild. Für jedes einzelne Pixel geschieht dann folgendes:

- Die zugehörige Farbe wird in ihre Rot, Grün- und Blaukomponenten umgerechnet. Dazu wird jeweils ein Fehlerwert addiert, der sich beim Zeichnen der vorhergehenden Pixel für dieses Pixel aufsummiert hat.

- Aus den 16 verfügbaren Farben wird

die ähnlichste durch Vergleich der Fehlerquadrate herausgesucht. Das ist der aufwendigste Teil des Programmes, die meiste Rechenzeit wird in dieser Schleife verpulvert!

- Die ähnlichste Farbe wird gespeichert, um später gezeichnet zu werden.

Nun stimmt diese Farbe mit der ursprünglichen Farbe des Pixels i.allg. nicht überein. Also entsteht beim Zeichnen ein Fehler, der ausgeglichen werden muß, indem er bei den folgenden Pixeln Berücksichtigung findet. Der Unterschied zwischen ursprünglicher und erzeugter Farbe wird - mit umgekehrtem Vorzeichen - auf die umliegenden Pixel verteilt. Das unmittel- bar rechts danebenliegende Pixel wird mit 2/8 bedacht, genauso das direkt darunterliegende. Das rechts darunterliegende Pixel erhält noch 1/8.

- Zuletzt werden die Pixel gezeichnet und die nächste Zeile in Arbeit genommen.

Zur Verteilung des Fehlers sind noch einige Anmerkungen zu machen:

Der Fehler wird nicht sofort durch 8 geteilt, sondern erst, wenn er zum Farbwert des Pixels hinzugerechnet wird. Dadurch wird zum einen eine kleine Zeitersparnis erzielt, zum anderen lassen sich Rundungsfehler vermindern, da mit einer Art Festkommaarithmetik gerechnet wird - nämlich 3 Binärstellen hinter dem Komma. Für den Fehler existieren - für jede Farbkomponente getrennt - Arrays, die die Fehler jeweils einer ganzen Zeile enthalten. Wenn ein Pixel bearbeitet wurde, enthält solch ein Array an dieser Stelle schon den Fehler für die nächste Zeile.

Wenn Sie den Algorithmus näher betrachten, stellen Sie fest, daß nur $^5/_8$, also 62.5% des gemachten Fehlers ausgeglichen werden. Diese Werte habe ich so festgelegt, da bei höheren Werten oftmals Fehler auftraten, die ich als "Überkompensationsfehler" oder auch als "Resonanzkatastrophen" bezeichnen könnte. Dadurch, daß ein Fehler zu gut ausgeglichen wurde - d.h. die ähnlichste Farbe bei Berücksichtigung des Fehlers war der ursprünglichen Farbe viel weniger ähnlich, als ohne Kompensation - wurde ein größerer Fehler erzeugt, wodurch dann nach einigen Iterationen der ganze Mechanismus entgleiste. In Folge zog sich über den restlichen Bildschirm eine Fahne von falschen Farben. (Ein überaus interessanter Defekt, aber nicht das gewünschte Ergebnis.) Sie können auch mit eigenen Werten Experimente anstellen - ich bin an Verbesserungen des Programmes immer interessiert, am meisten aber an Vorschlägen, wie die Suche nach der ähnlichsten Farbe noch weiter optimiert werden könnte.

Ein wenig Aufmerksamkeit möchte ich noch auf die Programmierung der Routinen zum Einlesen und Abspeichern der Bilder richten. Das Programm ist - bis auf die im ROM vorhandenen Fehler z.B. für die Fileselektorbox - kaum mehr zum Absturz zu bewegen, da man noch nicht einmal auf ein nicht existentes Laufwerk zugreifen kann!

Noch einige Anregungen und Anmerkungen:

Das Programm läßt sich auch noch für andere Zwecke mißbrauchen. Durch Wahl einer geeigneten Palette kann man z.B. farbige Bilder in Graustufen umrechnen lassen (aber in der 320*200-Auflösung!).

Zur Vorsicht möchte ich raten bei der Gestaltung eigener Paletten. Wenn Sie eine der Grundfarben - Rot, Grün und Blau - nicht genügend berücksichtigen, kann es leicht zu Overflow-Fehlern kommen, die evtl. zum Absturz führen.

Zur Zeit arbeite ich an einer Erweiterung des Programmes, um farbige Bilder möglichst exakt in ihr schwarzweißes Äquivalent umzuformen. Eine Erweiterung für die mittlere Auflösung halte ich dagegen nicht für sinnvoll. Diese wird auch kaum für grafische Zwecke verwendet - ich glaube, daß sie tatsächlich für überhaupt nichts verwendet wird.

Ich hoffe, daß ich Ihnen mit diesem Programm geholfen habe, die Grafikfähigkeiten des ATARIs voll auszuschöpfen, und wünsche Ihnen viel Erfolg bei eigenen Experimenten.

PP

```
 1:
 2:  #include <osbind.h>
 3:  #include <stdio.h>
 4:
 5:  /*
 6:    COLCONV.C  -  Version Nr. 1.021 A
 7:    Dieses Programm wandelt ein Bild (in der
      niedrigsten Auflösung)
 8:    mit einer beliebigen Farbpalette in eins mit
      einer festgelegten
 9:    Palette um. Dieses eignet sich dann für
      Überblendungen o.ä.
10:    Die Umwandlung erfolgt mittels eines selbst
      entwickelten Algorithmus,
11:    der die Abweichungen der Farben auf die
      umliegenden Punkte verteilt.
12:    Dabei wird mit einer Genauigkeit gearbeitet, die
      auch kleinere
13:    Fehler zuverlässig ausgleicht.
14:
15:    Hans-Detlef Siewert, Bruchbäumerweg 20 A,
          4780 Lippstadt
16:  */
17:
18:  /**********************************************/
19:   * Copyright by Hans-Detlef Siewert, HaDeS-Soft,
          April/Mai 1988 *
20:   **********************************************/
21:
22:  int  oldcolor[16], picolor[16];
23:  char Pathname[128], Filename[128];
24:  char *WS1, *WS2;
25:  char *Desktop, *WrkScr1, *WrkScr2;
```

```
26:  /*
27:   Desktop ist der Zeiger auf den normalen
      Bildschirmspeicher.
28:   Daneben werden zwei weitere Bildschirmspeicher
      angelegt. Der
29:   erste enthält das ursprüngliche Bild, der zweite
30:   dann das bearbeitete Bild.
31:  */
32:  int  Filehandle,Auflsg;
33:  int  rerr[321],gerr[321],berr[321];  /* Fehler für
                               Rot, Grün und Blau */
34:  int  col[321]; /* Farben einer Bildschirmzeile  */
35:  int  srpal[16],sgpal[16],sbpal[16];  /* Source-
                            Palette - Farbanteile  */
36:  int  trpal[16],tgpal[16],tbpal[16];  /* Target-
                            Palette - Farbanteile  */
37:  /*
38:   Das sind die Paletten, die gewählt werden können:
39:   (Sie können hier natürlich eigene Paletten
      verwenden)
40:  */
41:  int  paletten[2][16]={ { 0x0777, 0x0700, 0x0070,
        0x0770,
42:                   0x0007, 0x0707, 0x0077, 0x0555,
43:                   0x0333, 0x0733, 0x0373, 0x0773,
44:                   0x0337, 0x0737, 0x0377, 0x0000 },
45:                 { 0x0777, 0x0004, 0x0007, 0x0047,
46:                   0x0077, 0x0074, 0x0070, 0x0470,
47:                   0x0770, 0x0740, 0x0700, 0x0400,
48:                   0x0444, 0x0040, 0x0404, 0x0000 }};
49:  int  targpal;  /* Das ist die gewählte Palette  */
50:
51:
```

```
52:  /****
53:   Die folgenden Variablen und Routinen sind Teile
      eines von mir
54:   geschriebenen Bindings zur Benutzung der LineA-
      Grafikroutinen
55:   im ROM vom Megamax-C Compiler aus.
56:  ****/
57:
58:  /* Pointer auf den LineA- Parameterblock */
59:  static long LineA_PP;
60:
61:  lina_init()
62: /* Diese Routine initialisiert die LineA Routinen.*/
63:  { asm
64:     {
65:        dc.w   0xa000
           /* Aufruf der 'Initialization'- Routine */
66:        move.l A0,LineA_PP(A4) /* LineA_PP zeigt auf
                                  den Parameterblock */
67:     }
68:  }
69:
70:
71:  lina_setp(x,y,farbe)
72:/* Setzen eines einzelnen Punktes (x,y) mit'farbe'*/
73:  int x,y,farbe;
74:  { asm
75:     {
76:        move.l LineA_PP(A4),A0
           /* A0 zeigt auf den Parameterblock        */
77:        move.l 0x08(A0),A1
           /* A1 zeigt auf die int_in-Tabelle        */
78:        move.w farbe(A6),(A1)
           /* Farbwert nach int_in[0] übertragen     */
79:        move.l 0x0C(A0),A1
           /* A1 zeigt jetzt aufdie pts_in-Tabelle   */
80:        move.w x(A6),(A1)+     /* x nach pts_in[0]*/
81:        move.w y(A6),(A1)      /* y nach pts_in[1]*/
82:        dc.w   0xa001
           /* Aufruf von 'Put_Pixel'                 */
83:     }
84:  }
85:
86:
87:  /* Routinen zum Abspeichern und Laden von Bildern
        im DEGAS-Format */
88:
89:  int Screen_Save()
90:  /* Speichert Auflösung, Palette und Bild-Daten im
        DEGAS-Format    */
91:  {
92:    long Length, Count;
93:
94:    Filehandle= Fcreate (Filename,0);
95:    if (Filehandle<=0)                /* Fehler beim
                                            Anlegen */
96:      return(0);
97:    Length= 2L;          /* Auflösung schreiben */
98:    Count = Fwrite (Filehandle, Length, (char *)
            &Auflsg);
99:    Length= 32L;  /* Farbpalette abspeichern */
100:   Count+= Fwrite (Filehandle, Length, (char *)
            &picolor[0]);
101:   Length= 32000L;  /* Bild-Daten abspeichern */
102:   Count+= Fwrite (Filehandle, Length, (char *)
            WrkScr2);
103:   Fclose (Filehandle);
104:   return (Count==32034);
105: }
106:
107: int Screen_Load()
108: /* Lädt Bild im DEGAS-Format in die Workscreen1 */
109: {
110:   long Length, Count;
111:
112:   Filehandle=Fopen (Filename,0);
113:   if (Filehandle<=0) /* Fehler beim Öffnen */
114:     return(0);
115:   Length= 2L;          /* Auflösung lesen */
116:   Count = Fread (Filehandle, Length, (char *)
            &Auflsg);
117:   Length= 32L;  /* Farbenpalette lesen:        */
118:   Count+= Fread (Filehandle, Length, (char *)
            &picolor[0]);
119:   Length= 32000L;  /* Bild-Daten lesen */
120:     Count+= Fread (Filehandle, Length, (char *)
             WrkScr1);
121:     Fclose (Filehandle);
122:     return (Count==32034);
123:  }
124:
125:  int Get_Memory()
126:  /* Initialisiert Variablen und legt
        Arbeitsbildschirme an.        */
127:  {
128:    register int i;
129:    unsigned amount,size;
130:
131:    /* Arbeitsbildschirme:       */
132:    amount= 8064;
133:    size=4;
134:    WS1= (char *) calloc(amount,size);
135:    if (!WS1)      /* War genug Speicher da?      */
136:      return(0);
137:    WS2= (char *) calloc(amount,size);
138:    if (!WS2)      /* War genug Speicher da?      */
139:      return(0);
140:    /* Seitengrenzen einhalten: */
141:    WrkScr1= (char *) ( (long)WS1 & 0xFFFFFF00L);
142:    WrkScr2= (char *) ( (long)WS2 & 0xFFFFFF00L);
143:    Pathname[0]= '\0'; /* Strings initialisieren */
144:    Filename[0]= '\0';
145:    for (i=0; i<16; i++)     /* Alte Farben merken */
146:      oldcolor[i]= Setcolor(i,-1);
147:    /* Fehlertabellen löschen    */
148:    for (i=0; i<320; i++)
149:      { rerr[i]=0; gerr[i]=0; berr[i]=0; }
150:    return(1);
151:  }
152:
153:  Init_Path()
154:  /* Ermittelt aktuellen Pfadnamen für die
        Fileselektor-Box        */
155:  {
156:    char Path[128];
157:
158:    /* Aktuelles Laufwerk:       */
159:    sprintf (Pathname,"%c:",(char) 65+ Dgetdrv());
160:    Dgetpath (Path,0); /* Aktuelles Verzeichnis */
161:    strcat (Pathname, Path);
162:    strcat (Pathname, "\\*.PI1");
163:  }
164:
165:  int Ask_Fname()
166:  /* Fragt mit einer Fileselektorbox nach dem
        Filenamen,            */
167:  /* setzt das gewünschte Laufwerk und den
        gewünschten Pfad        */
168:  /* und fängt nebenbei alle möglichen Fehler ab. */
169:  {
170:    int Button,DDrive;
171:    long ADrives;      /* Angemeldete Laufwerke */
172:    char Pfad[128], *sptr1, *sptr2;
173:
174:    strcpy (Pfad, Pathname);
175:    /* File-Selektorbox für Frage nach Filenamen:*/
176:    fsel_input (Pfad, Filename, &Button);
177:    if (!Button)      /* Abbruch-Button betätigt?  */
178:      return(0);
179:    if (!strlen(Filename))/* Kein File angewählt?*/
180:      return(0);
181:    DDrive= (int)Pfad[0]-65;     /*Gewünschtes
                                       Laufwerk? */
182:    ADrives= Drvmap(); /* Laufwerk angeschlossen? */
183:    if (!( (1<<DDrive) & ADrives))
184:      return(0);
185:    Dsetdrv(DDrive);  /* Laufwerk setzen       */
186:                      /* Pfadnamen kürzen:     */
187:    sptr1= &Pfad[2];  /*  Laufwerksbez. löschen: */
188:    sptr2= Pfad;
189:    while (*sptr1!='\0') *sptr2++= *sptr1++;
190:    *(++sptr2)= '\0';
191:    sptr1= &Pfad[ strlen(Pfad) ]; /* Wildcard
                                        löschen */
192:    while (*sptr1 != '\\') *sptr1--= '\0';
193:    if (Dsetpath(Pfad)<0) /* Pfadnamen setzen */
194:      return(0);
195:    return(1);
196:  }
197:
198:  int Read_Pic()
```

```
199:  /* Fragt nach dem Namen eines Bildes und liest
        Workscreen1 ein.                                  */
200:  {
201:    int i,ok;
202:
203:    if (!Ask_Fname()) /* Filename abfragen         */
204:      return(0);
205:    graf_mouse(256,0L);     /* Maus abschalten    */
206:    Setscreen(WrkScr1, WrkScr1, -1); /* Workscreen1
                                            anzeigen */
207:    ok= Screen_Load();            /* Bild laden*/
208:    Setscreen(Desktop, Desktop, -1);/* Desktop
                                           anzeigen  */
209:    graf_mouse(257,0L);     /* Maus einschalten   */
210:    if (!ok)                /* Fehler beim Laden? */
211:      return(0);
212:    return (Auflsg==0);/* Ist es ein Lowrez-Bild?*/
213:  }
214:
215:  int Write_Pic()
216:  /* Fragt nach dem Namen und speichert
        Workscreen2 ab.                                   */
217:  {
218:    int i,ok;
219:
220:    if (!Ask_Fname())     /* Filename abfragen      */
221:      return(0);
222:    Auflsg= 0;         /* niedrige Auflösung        */
223:    for (i=0; i<16; i++) /* Farbwerte holen         */
224:      picolor[i]= paletten[targpal][i];
225:    graf_mouse(256,0L);  /* Maus ausschalten        */
226:    Setscreen(WrkScr2, WrkScr2, -1); /* Workscreen2
                                            anzeigen */
227:    ok= Screen_Save();    /* Bild abspeichern       */
228:    Setscreen(Desktop, Desktop, -1); /* Desktop
                                            anzeigen */
229:    graf_mouse(257,0L);     /* Maus einschalten:  */
230:    return(ok);
231:  }
232:
233:  Clean_up()
234:  /* Gibt reservierten Speicher frei und restauriert
        die Farben     */
235:  {
236:    register int i;
237:
238:    free(WS1);
239:    free(WS2);
240:    for (i=0; i<16; i++)
241:      Setcolor(i, oldcolor[i]);
242:  }
243:
244:  getrow(line_addr,tab_addr)
245:  register char *line_addr, *tab_addr;
246:  /*
247:   Diese Assemblerroutine wandelt eine komplette
       Bildschirmzeile
248:   in die einzelnen Farbwerte um. line_addr zeigt
       auf die Start-
249:   adresse der Bildschirmzeile, tab_addr auf die
       Startadresse der
250:   Tabelle, in der die Farbwerte gespeichert werden
       sollen.
251:   Die Verarbeitung geschieht wortweise. Eine äußere
       Schleife wird
252:   20 mal ausgeführt, dort werden jeweils immer 4
       neue Worte geholt
253:   (eines für jedes Farbplane). Diese Worte werden
       dann in einer
254:   inneren Schleife in die Werte zerlegt, dadurch
       erhält man 16
255:   Farbwerte. Insgesamt werden also 20*16=320
       Pixelwerte erzeugt.
256:  */
257:
258:  { asm
259:      {
260:          movem.l  D0-D6,-(A7)  /* Register retten */
261:          moveq     #0x13,D0
            /* D0 zählt die Worte pro Farbplane*/
262:        w_loop:      /* ÄuPere Schleife, 20*        */
263:          move.w    (line_addr)+,D2
           /* D2 enthält  Farbplane #0             */
264:          move.w    (line_addr)+,D3
           /* D3 enthält  Farbplane #1             */
265:          move.w    (line_addr)+,D4
           /* D4 enthält  Farbplane #2             */
266:          move.w    (line_addr)+,D5
           /* D5 enthält   Farbplane #3            */
267:          moveq     #0x0F,D1
           /* D1 zählt die   Bits pro Wort         */
268:        b_loop:  /* Innere Schleife, 16*            */
269:          moveq     #0x00,D6            /* D6 löschen */
270:          roxl.w    #1,D5
           /* höchstwertiges Bit holen              */
271:          roxl.w    #1,D6  /*  ... und speichern    */
272:          roxl.w    #1,D4  /* zweithöchstes Bit holen*/
273:          roxl.w    #1,D6  /*  ... und speichern    */
274:          roxl.w    #1,D3  /* dritthöchstes Bit holen*/
275:          roxl.w    #1,D6  /*  ... und speichern    */
276:          roxl.w    #1,D2
           /* niederwertigstes Bit holen         */
277:          roxl.w    #1,D6  /*  ... und speichern    */
278:          move.w    D6,(tab_addr)+ /* Farbwert in
                                  Tabelle speichern */
279:          dbf       D1,b_loop    /* Ende der
                                   inneren Schleife */
280:          dbf       D0,w_loop    /* Ende der
                                   äußeren Schleife */
281:          movem.l  (A7)+,D0-D6  /* Alte Registerwerte
                                   holen            */
282:      }
283:  }
284:
285:  int Picture_Work()
286:  /* Bearbeitet das geladene Bild.                  */
287:  {288:    int i, x, y;/* Schleifen-Laufvariablen */
289:    int r,g,b;              /* Farbanteile  */
290:    int Abweichung, MinAbw;
291:    int nr,ng,nb;  /* neu gemachter Fehler       */
292:    int fr,fg,fb;  /* Fehler in den Farbanteilen */
293:    register int scol;  /* Ursprüngliche Farbe    */
294:    register int *rval, *gval, *bval;
         /* Farben-Vergleichswerte        */
295:    register int *color;  /* Zeiger auf
                                 Farbentabelle    */
296:
297:    graf_mouse(256,0L);    /* Maus ausschalten     */
298:    Setscreen(WrkScr2, WrkScr2,-1); /* Workscreen2
                                           anzeigen */
299:    for (i=0; i<16; i++) /* Neue Palette setzen    */
300:      Setcolor(i, paletten[targpal][i]);
301:  /* Ausgangs- und Zielpaletten in ihre
        Farbanteile umrechnen                           */
302:    for (i=0; i<16; i++)
303:    {
304:      srpal[i]= (picolor[i] & 0x0700)>>8;
305:      sgpal[i]= (picolor[i] & 0x0070)>>4;
306:      sbpal[i]=  picolor[i] & 0x0007;
307:      trpal[i]= (paletten[targpal][i] & 0x0700)>>8;
308:      tgpal[i]= (paletten[targpal][i] & 0x0070)>>4;
309:      tbpal[i]=  paletten[targpal][i] & 0x0007;
310:    }
311: /* Berechnungsssschleife f.alle Bildschirmzeilen */
312:    for (y=0; y<200; y++)
313:      {
314:      /* Farben in einer Zeile lesen                */
315:        getrow (WrkScr1 + y*160, col);
316:      /* noch keine Abweichungen in dieser Zeile */
317:        nr=0; ng=0; nb=0;
318:        /* Berechnung der neuen Farben              */
319:        color= &col[0];
320:        for (x=0; x<320; x++)
321:        {
322:          scol= *color;
323:          r= srpal[scol]+ (rerr[x]>>3);
324:          g= sgpal[scol]+ (gerr[x]>>3);
325:          b= sbpal[scol]+ (berr[x]>>3);
326:          MinAbw= 255;
327:          rval= &trpal[0];
328:          gval= &tgpal[0];
329:          bval= &tbpal[0];
330:          /* ähnlichste Farbe suchen                */
331:          for (i=0; i<16;i++)
332:          {
333:            /* Rot-Anteil                           */
334:            Abweichung = (*rval - r) *(*(rval++)-r);
335:            /* Grün-Anteil        */
336:            Abweichung+= (*gval - g) *(*(gval++)-g);
337:            /* Blau-Anteil       */
338:            Abweichung+= (*bval - b) *(*(bval++)-b);
339:            if (Abweichung<MinAbw)
340:              { scol=i; MinAbw= Abweichung; }
```

```
341:            }
342:            /* Nächster Punkt      */
343:            *(color++)= scol;
344:            /* gemachten Fehler merken */
345:            fr= r- trpal[scol];
346:            fg= g- tgpal[scol];
347:            fb= b- tbpal[scol];
348:            /* Fehler auf nachfolg.Punkte verteilen */
349:            rerr[x+1]+= fr<<1;
350:            gerr[x+1]+= fg<<1;
351:            berr[x+1]+= fb<<1;
352:            rerr[x]= nr+ fr<<1;
353:            gerr[x]= ng+ fg<<1;
354:            berr[x]= nb+ fb<<1;
355:            nr= fr;
356:            ng= fg;
357:            nb= fb;
358:          }
359:          /* Plotten der neuen Farben        */
360:          color= &col[0];
361:          for (x=0; x<320; x++)
362:            lina_setp(x, y, *(color++));
363:        }
364:    /* Neues Bild fertiggestellt, Bildschirm
           zurückschalten.         */
365:    for (i=0; i<16; i++)  /* Alte Palette setzen */
366:      Setcolor(i, oldcolor[i]);
367:    Setscreen(Desktop, Desktop, -1);
          /* Desktop anzeigen             */
368:    graf_mouse(257,0L);     /* Maus einschalten   */
369:    return(1);
370:  }
371:
372:  main()
373:  /***** Hauptprogramm *****/
374:  {
375:    appl_init();
376:    lina_init();
        /* Initialisierung der LineA-Routinen.    */
377:    while (Cconis())
378:      Crawcin();
379:
380:    Desktop= (char *) Physbase();
381:    if (Getrez())     /* Nicht niedrigste Auflösung */
382:    {
383:      form_alert(1,"[1][Geht nur|in LowRes!][
          Ach was  ? ]");
384:      exit(0);
385:    }
386:
387:    if (!Get_Memory())
388:  /* Speicher anfordern hat nicht hingehauen.   */
389:  /* Entweder Sie verkleinern Ihre RAM-Disk oder*/
390:    /* Sie verkleinern Ihr Vermögen...          */
391:    {
392:      form_alert(1,"[1][Zu wenig|Speicher!]
                   [ Tja...]");
393:      exit(0);
394:    }
395:
396:    Init_Path();
397:
398:    if (!Read_Pic()) /* Einlesen des Bildes      */
399:    /* Dabei kann *alles* mögliche passieren.   */
400:    {
401:      Clean_up();
402:      form_alert(1,"[3][Lesefehler!][ Mist ]");
403:      exit(0);
404:    }
405:
406:    /*
407:     Frage nach der Farbpalette, die für das
         umgewandelte
408:     Bild verwendet werden soll. Leider können die
         Alert-
409:     boxen in der Low-Resolution nicht allzu groß
         sein, so
410:     daß man sich sehr knapp fassen muß.
411:     Palette 1 ist die Standard-Farbenpalette des
         Desktops.
412:     Palette 2 ist eine regenbogenähnliche Palette.
413:    */
414:    targpal= form_alert(1,"[2][Welche|Palette?]
             [1|2]")-1;
415:
416:    if (!Picture_Work())
        /* Bearbeitung des Bildes      */
417:    /* Solch ein Fehler darf eigentlich nicht
          geschehen! */
418:    {
419:      Setscreen(Desktop, Desktop, -1);
420:      Clean_up();
421:      form_alert(1,"[2][Interner Fehler!][ Pech ]");
422:      exit(0);
423:    }
424:
425:    if (!Write_Pic())      /* Bild speichern      */
426:  /* Schreibschutz, kein Platz, defekte Diskette?*/
427:      form_alert(1,"[3][Schreibfehler!][ Schade ]");
428:
429:    Clean_up();
430:    appl_exit();
431:  }
```

Listing: Das Umwandlungsprogramm

ENDE

ASCII-Datei-Formatierer

Dietmar Rabich

Oft besteht der Wunsch, eine ASCII-Datei nachträglich zu bearbeiten, etwa durch das Hervorheben einiger besonderer Worte. Mit diesem Formatierer ist dies für alle ASCII-Texte möglich.

Nun, was kann der Formatierer? Mit ihm ist es möglich, bestimmte Worte (hier als Schlüsselworte bezeichnet) durch Großschreibung hervorzuheben. Zusätzlich kann man Zeilennummern vor jede Zeile schreiben.

Und wie arbeitet man mit dem Programm? Ganz einfach. Mit irgendeinem ASCII-Editor fertigt man sich eine Schlüsselwortdatei an. Pro Zeile steht dort ein bis zu 40 Zeichen langes Wort. (*'Donaudampfschifffahrtsgesellschaftskapitän'* geht leider nicht mehr!) Wichtig ist, daß man eventuell Leerzeichen hinzufügt. Untersucht man etwa einen PASCAL-Quelltext auf das Wort for, so werden alle Buchstabenkombinationen mit for hervorgehoben, also auch ein Name wie format. Dieser wird zu FORmat. Hängt man an das *'for'* ein Leerzeichen, kann diese *Fehlformatierung* nicht auftreten. Hat man eine solche Schlüsselwort-Datei zur Verfügung, braucht man nur noch das Programm *FORMATER.TTP* aufzurufen und in der Dialogbox erst die zu verändernde Datei, dann die Schlüsselwortdatei und schließlich eventuell ein '+' für Zeilennummern einzutragen.

Die Schlüsselwortdateien haben einen großen Vorteil gegenüber steifen Textformatierern. Man kann beliebige ASCII-Texte bearbeiten (etwa Sourcecodetexte in BASIC, PASCAL oder anderen Programmiersprachen) und dort bestimmte Worte hervorheben. Beispielsweise könnte man in einem PASCAL-Sourcecode Pascal- oder Library-Schlüsselworte hervorheben.

Nun etwas zum Programm. Das in ST Pascal Plus geschriebene Programm liest erst eine Schlüsselwort- und dann eine Textdatei zeilenweise ein. Beim Einlesen der Schlüsselworte werden diese automatisch in Kleinbuchstabenschreibweise übersetzt. Also braucht man auf die Schreibweise der Schlüsselworte nicht unbedingt zu achten. Jede Textzeile wird nacheinander nach jedem Schlüsselwort untersucht. Hierzu wird jeweils eine Kennzeile erzeugt (mit Kleinbuchstaben), in der die ebenso klein geschriebenen Schlüsselworte gesucht werden. Sollte ein Schlüsselwort gefunden werden, wird es in der Text- und der Kennzeile groß geschrieben. Somit kann es in der Kennzeile zumindest an dieser Stelle nicht erneut gesucht werden. Die fertige Zeile wird ggf. mit Zeilennummer in die neue Datei *OUTPUT.TXT* geschrieben. Für die PASCAL-Programmierer sind noch einige Bemerkungen im Listing zu finden. Das Programm wurde als TOS-Anwendung übersetzt und nachträglich von .TOS in .TTP umbenannt.

Als Beispiel einer solchen Textformatierung wurde das Listing dieses Programmes bearbeitet.

```
 1:  (***********************************************)
 2:  (*      Universal-Formatierer für ASCII-Texte    *)
 3:  (* ------------------------------------------- *)
 4:  (* Mit  Hilfe dieses  kleinen  Programmes  können
     bestimmte  Worte in *)
 5:  (* beliebigen   ASCII-Texten   durch
     Großschreibung   dieser  Worte *)
 6:  (* hervorgehoben  werden.  Besonders  eignet  sich
     dieses   Programm, *)
 7:  (* um  Programmtexte   in  BASIC,  PASCAL   oder
     anderen  Programmier- *)
 8:  (* sprachen zu bearbeiten.                       *)
 9:
10:  (* Autor                : Dietmar Rabich,
     Dövelingsweg 2, 4408 Dülmen                      *)
11:  (* Programmiersprache : ST Pascal Plus
     Version 1.20                                     *)
12:  (* Version              : 1.0                     *)
13:  (* Datum                : 14. September 1987      *)
14:  (***********************************************)
15:
16:  program ascii_formater;
17:
18:  const max_worte = 250; (* maximale Anzahl der
                              Schlüsselworte *)
19:
20:  type str255 = string[255];  (* maximaler String,
                                    255 Zeichen     *)
21:       str40  = string[40];   (* Schlüsselwort,
22:                                 maximal 40 Zeichen *)
23:  var anz_worte : integer; (* Anzahl der
                                 Schlüsselworte      *)
24:      wort      : array [1..max_worte] of str40;
                             (* Array für            *)
```

```
25:                        (* Schlüsselworte        *)
26:
27: (* Procedure,die die Schlüsselworte aus Datei liest *)
28:  procedure schluessel;
29:
30:   var datei : text;     (* Textdatei     *)
31:       name  : string;   (* Dateiname     *)
32:       i     : integer;  (* Laufvariable  *)
33:
34: (* Wandelt Schlüsselworte in Worte mit
     Kleinbuchstaben um *)
35:   procedure wandle40(var s:str40);
36:
37:    var i : integer;
38:
39:    begin
40:     for i:=1 to length(s) do
41:      if s[i] in ['A'..'Z'] then
42:       s[i]:=chr(ord(s[i])-ord('A')+ord('a'))
43:      else
44:       case s[i] of
45:        'Ä' : s[i]:='ä';(*Umlaute extra betrachtet*)
46:        'Ö' : s[i]:='ö';
47:        'Ü' : s[i]:='ü';
48:       end;
49:    end;
50:
51:   begin
52:
53:    (* Namen der Schlüsselwortdatei ermitteln *)
54:    name:='';
55:    cmd_getarg(2,name); (* Liest ggf. Dateinamen aus
                           Command-Line *)
56:    if name='' then
57:      begin
58:       write('Bitte Schlüsselwortdateiname:');
59:       readln(name);
60:      end
61:     else
62:      writeln('Schlüsselwortdatei: ',name);
63:
64:    (* Schlüsselwortdatei einlesen *)
65:    i:=0;
66:    reset(datei,name);
67:    while (not(eof(datei)) and (i<max_worte)) do
68:     begin
69:      i:=i+1;
70:      readln(datei,wort[i]);
71:      wandle40(wort[i]);
72:     end;
73:    close(datei);
74:    anz_worte:=i;
75:   end;
76:
77:  (* Procedure, um Zeile zu formatieren *)
78:  procedure format(var z:str255);
79:
80:   var i    : integer; (* Laufvariable           *)
81:       p    : integer; (* Position Schlüsselwort *)
82:       kenn : str255;  (* Kennzeile              *)
83:
84:   (* Kennzeile (mit Kleinbuchstaben) erzeugen *)
85:   procedure erzeuge_kennung(z:str255;var k:str255);
86:   (* Procedure ähnlich 'wandle40, aber mit str255*)
87:
88:    var i : integer;
89:
90:    begin
91:     k:=z;
92:     for i:=1 to length(k) do
93:      if k[i] in ['A'..'Z'] then
94:       k[i]:=chr(ord(k[i])-ord('A')+ord('a'))
95:      else
96:       case k[i] of
97:        'Ä' : k[i]:='ä';(*Umlaute extra betrachtet*)
98:        'Ö' : k[i]:='ö';
99:        'Ü' : k[i]:='ü';
100:       end;
101:    end;
102:
103:   (* Zeichen im String s hervorheben *)
104:   procedure ersetze(var s:str255; m,n:integer);
105:
106:    var i : integer;
107:
108:    begin
109:     for i:=m to m+n-1 do
110:      if s[i] in ['a'..'z'] then
111:       s[i]:=chr(ord(s[i])-ord('a')+ord('A'))
112:      else
113:       case s[i] of
114:        'ä' : s[i]:='Ä';
              (* Umlaute extra betrachtet *)
115:        'ö' : s[i]:='Ö';
116:        'ü' : s[i]:='Ü';
117:       end;
118:    end;
119:
120:   begin
121:    for i:=1 to anz_worte do
122:     begin
123:      erzeuge_kennung(z,kenn);(*Kennung für jedes *)
124:      p:=pos(wort[i],kenn);   (* Schlüssel wort    *)
125:      while ((p>=1) and (p<=length(z))) do
126:       begin
127:        ersetze(z,p,length(wort[i]));
              (* Wort in Zeile hervorheben *)
128:        ersetze(kenn,p,length(wort[i])); (* Ebenso
                               in der Kennzeile   *)
129:        p:=pos(wort[i],kenn);
130:       end;
131:     end;
132:   end;
133:
134:  (* ASCII-Textdatei ändern *)
135:  procedure aendere_text;
136:
137:   var datei,datei2 : text; (* Textdateien        *)
138:       name         : string;  (* Dateiname       *)
139:       zeile        : str255;  (* Textzeile       *)
140:       zlnr         : string;
                        (* Zeilennummerstatus*)
141:       nr           : integer; (* Zeilennummer    *)
142:
143:   begin
144:
145:    (* Dateinamen ermitteln *)
146:    name:='';
147:    cmd_getarg(1,name); (* Liest ggf. Dateinamen aus
           Command-Line *)
148:    if name='' then
149:      begin
150:       write('Bitte Dateiname:');
151:       readln(name);
152:      end
153:     else
154:      writeln('Eingabedatei: ',name);
155:
156:    cmd_getarg(3,zlnr);  (* Liest ggf. Zeichen aus
                           Command-Line *)
157:
158:    (* Datei verändern, neue Datei: OUTPUT.TXT *)
159:    writeln('Ausgabedatei: OUTPUT.TXT');
160:    nr:=0;
161:    reset(datei,name);
162:    rewrite(datei2,'OUTPUT.TXT');
163:    while not(eof(datei)) do
164:     begin
165:      nr:=nr+1;          (* aktuelle Zeilennummer *)
166:      readln(datei,zeile);
167:      format(zeile);
168:      if zlnr='+' then
169:       write(datei2,'{ ',nr:5,': } ');
                   (* Zeilennummer hinzufügen *)
170:      writeln(datei2,zeile);
171:     end;
172:    close(datei2);
173:    close(datei);
174:   end;
175:
176:  (* Hauptprogramm *)
177:  begin
178:   writeln('Universalformatierer für ASCII-Texte
            ((p)1987/ D. Rabich)');
179:   writeln('<<<<<<<<<<< Entwickelt mit ST Pascal
                Plus 1.20 >>>>>>>>>>>');
180:   schluessel;      (* Schlüsselworte lesen       *)
181:   aendere_text;    (* Text einlesen, ändern und
                        ausgeben *)
182:  end.
183:
184:  (*....  P r o g r a m m e n d e  ............*)
```

Listing: Universal-Formatierer für ASCII-Texte

Installieren von STAD-Fonts

Heinz Katzenmayer

Das neue GFA-BASIC 3.0 hat nicht nur eine Anzahl neuer Befehle, sondern auch Erweiterungen bereits vorhandener. Eine dieser Befehlserweiterungen ist der Parameter Fonts im Befehl DEFTEXT. Einziger Hinweis im Handbuch lautet: Installieren zusätzlicher Zeichensätze unter GDOS mit VST_LOADFONT, VQT-NAME ...). Da ich nicht über GDOS verfüge, war die Frage: Was gibt es noch für eine Möglichkeit, um diesen neuen Parameter zu nutzen? Ich erinnerte mich an einen Artikel in der ST Computer 4/88 "Schönere Programme durch Zeichensätze". In diesem Artikel wird eine Lösung in ST-Pascal Plus zu diesem Problem vorgestellt. Nach zusätzlicher Lektüre der Seiten 274 und 275 des Buches ATARI ST Intern, auf denen der Aufbau des Fontheaders beschrieben ist, machte ich mich an die Arbeit.

Als erstes benötigt man natürlich zusätzliche Fonts. Ich wählte den gleichen Weg, wie in dem vorgenannten Artikel aufgezeigt. Das Programm *STAD:GEM.GFA* lädt STAD-Fonts, ändert sie auf GEM-Format und legt sie unter der Kennung *GEM_F2* bis *-8* im Ordner *GEMFONTS* ab. Das Format der Fontdaten muß geändert werden, da bei STAD die 16 Bytes jedes Zeichens hintereinander liegen, dann kommen 16 Bytes des nächsten Zeichens usw., und das 256mal. GEM benötigt die Daten aber folgendermaßen: Das 1. Byte des 1. Zeichens, dann das 1. Byte des 2. Zeichens usw. Folglich ist das 257. Byte das 2. Byte des 1. Zeichens (bei 256 Zeichen). Verstanden? Das klingt komplizierter, als es ist (siehe Programmzeilen 40-47). Zusätzlich muß ein Fontheader für jedes Font angelegt werden. Allerdings muß bei einem neuen Font nicht der gesamte Header geändert werden. Dies ist nur an den fünf folgenden Stellen nötig:

1. Bytes 0+1, Font-Identifkations-Nummer (hier 2 bis 8, 1 ist Systemfont).
2. Bytes 4-35, Name des Fonts (Fontname von STAD plus Zusatz ".STAD"
3. Bytes 76-79, Adresse der Fontdaten.
4. Bytes 84-87, Adresse des nächsten Fontheaders.
5. Bytes 66+67, Flag zur Kennung, ob Font Systemfont ist.

Deshalb werden nur Fontnummer und -name zusammen mit den Fontdaten abgespeichert. Damit ist die Aufgabe des Programmes STAD:GEM.GFA erledigt. Im zweiten Programm, F_LADEN.GFA, werden nun diese Daten eingelesen. Was dann noch passiert, ist ziemlich einfach. Der Fontheader wird an den oben genannten Stellen geändert und dann zusammen mit den Fontdaten an einem vorher reservierten Speicherplatz abgelegt. Es wird also eine Fonttabelle an die nächste angehängt. In der letzen Tabelle wird bei der Adresse des nächsten Fontheaders (siehe 4.) keine neue Adresse mehr eingetragen. Alles, was sonst noch im Programm F_LADEN.GFA steht, wird nur zum Testen benutzt. Um das Programm nutzen zu können, muß man es als Unterprogramm einbauen. Siehe hier das Programm PROCFONT. GFA. Wenn man nicht alle sieben zusätzlichen Fonts im Programm braucht, muß man die Zählschleife i% nur entsprechend verändern

Natürlich kann man nur bei GEM-Anwendungen diese Fonts nutzen (im GFA-BASIC beim Befehl TEXT, nicht bei PRINT).

Nun wünsche ich viel Spaß und auch etwas Fingerspitzengefühl beim Nutzen der Fonts in Ihren Programmen. Als gelernter Schriftsetzer stehen mir öfters die Haare zu Berge, wenn ich die Verwendung von Schriften und Schriftstilen in Computerdrucksachen und Programmen sehe. Mit Desktop-Publishing werden diesem Horror jetzt fast unbeschränkte Möglichkeiten eröffnet.

PP

```
  1:  ' F_LADEN.GFA
  2:  ' PROGRAMM ZUM INSTALLIEREN DER FONTS
  3:  ' HEINZ KATZENMAYER, 10/88
  4:  ' BENOETIGT WERDEN NUR DIE ZEILEN:
  5:  ' 12-15, 21-62, 115-120
  6:  '
  7:  ' Textzeilen fuer Procedure AUSGABE:
  8:  test1$="ABCDEFGHIJKLMNOPQRSTUVWXYZ "
  9:  test2$="abcdefghijklmnopqrstuvwxyz "
 10:  test3$="012345679 äöüP !.,:;-"
 11:  test$=test1$+test2$+test3$
 12:  '
 13:  ' Neue Font- und Headeradressen:
 14:  DIM newfont_adr%(6),newheader_adr%(6)
 15:  header_adr%={L~A-906}
 16:  font_adr%=LPEEK({L~A-906}+76)
 17:  '
 18:  CLS
 19:  PRINT AT(10,7);"Fontdaten werden geladen:"
 20:  PRINT AT(10,9);"Font 1: Systemfont"
 21:  '
 22:  ' Fonts 2 bis 8 installieren:
 23:  FOR i%=0 TO 6
 24:    ' Speicherplatz fuer Header- und Fontdaten:
 25:    newfont$="Font"+STR$(i%)+SPACE$(4091)
 26:    newheader$="Header"+STR$(i%)+SPACE$(81)
 27:    nr$=SPACE$(2)
 28:    fname$=SPACE$(32)
 29:    newfont_adr%(i%)=VARPTR(newfont$)
 30:    newheader_adr%(i%)=VARPTR(newheader$)
 31:    nr_adr%=VARPTR(nr$)
 32:    fname_adr%=VARPTR(fname$)
 33:    '
 34:    ' Font- und Headerdaten laden:
 35:    f$="\GEMFONTS\GEM_F"+STR$(i%+2)+".FNT"
 36:    OPEN "I",#1,f$
 37:    BGET #1,nr_adr%,2
 38:    BGET #1,fname_adr%,32
 39:    BGET #1,newfont_adr%(i%),4096
 40:    CLOSE #1
 41:    '
 42:    ' Am Ende des Fontheaders Adresse des naechsten
        Headers:
 43:    LPOKE header_adr%+84,newheader_adr%(i%)
 44:    '
 45:    ' Header kopieren fuer neuen Header:
 46:    BMOVE header_adr%,newheader_adr%(i%),88
 47:    '
 48:    ' Header aendern (Nr., Name, Datenadr., Bit 0 im
        Flag)
 49:    DPOKE newheader_adr%(i%),DPEEK(nr_adr%)
 50:    BMOVE fname_adr%,newheader_adr%(i%)+4,32
 51:    LPOKE newheader_adr%(i%)+76,newfont_adr%(i%)
 52:    DPOKE newheader_adr%(i%)+66,12
 53:    '
 54:    ' Aus neuer Adresse wird alte (fuer Zeile 44):
 55:    header_adr%=newheader_adr%(i%)
 56:    '
 57:    ' Font ist geladen:
 58:    GOSUB name
 59:    PRINT AT(10,10+i%);"Font ";i%+2;": ";n$
 60:  NEXT i%
 61:  '
 62:  ' Beim letzten Fontheader kein weiterer Header:
 63:  LPOKE header_adr%+84,0
 64:  '
 65:  ' Ab hier brauchen man das Programm spaeter nicht
      mehr:
 66:  ' Das ist nur der Test:
 67:  '
 68:  PAUSE 100
 69:  CLS
 70:  GOSUB ausgabe
 71:  PAUSE 100
 72:  al$="Wollen Sie noch|ein Muster von|jedem Font
         sehen"
 73:  ALERT 2,al$,1," JA |NEIN",al%
 74:  '
 75:  IF al%<>1
 76:    END
 77:  ENDIF
 78:  '
 79:  DO
 80:    REPEAT
 81:      CLS
 82:      PRINT AT(2,2);"Welches Font (1-8 / 0=Ende):";
 83:      PRINT AT(32,2);
 84:      FORM INPUT 1,x$
 85:    UNTIL ASC(x$)>47 AND ASC(x$)<57
 86:    '
 87:    font%=VAL(x$)
 88:    EXIT IF font%=0
 89:    FOR attr%=0 TO 5
 90:      DEFTEXT 1,2^attr%,,13,font%
 91:      TEXT 10,40+((attr%+1)*30),test$
 92:    NEXT attr%
 93:    DEFTEXT 1,0,,32
 94:    TEXT 10,280,test1$
 95:    TEXT 10,330,test2$
 96:    TEXT 10,380,test3$
 97:    '
 98:    DEFTEXT 1,0,,6,1
 99:    TEXT 480,370,"Drücken Sie eine"
100:    TEXT 480,380,"beliebige Taste!"
101:    VOID INP(2)
102:  LOOP
103:  END
104:  '
105:  PROCEDURE ausgabe
106:    FOR n%=1 TO 8
107:      DEFTEXT ,,,13,1
108:      y%=-25+n%*48
109:      TEXT 10,y%,"Font Nummer "+STR$(n%)+":"
110:      DEFTEXT ,,,13,n%
111:      ADD y%,20
112:      TEXT 10,y%,test$
113:    NEXT n%
114:  RETURN
115:  '
116:  PROCEDURE name
117:    CLR n$
118:    FOR n%=0 TO 31
119:      n$=n$+CHR$(PEEK(fname_adr%+n%))
120:    NEXT n%
121:  RETURN
```

Listing 1: F_Laden. GFA

```
  1:  ' PROCEDURE FONT
  2:  ' ZUM EINBINDEN IN EIGENE PROGRAMME
  3:  ' HEINZ KATZENMAYER, 10/88
  4:  '
  5:  PROCEDURE font
  6:    LOCAL i%,f$
  7:    DIM newfont_adr%(6),newheader_adr%(6)
  8:    header_adr%={L~A-906}
  9:    font_adr%=LPEEK({L~A-906}+76)
 10:    FOR i%=0 TO 6
 11:      newfont$="Font"+STR$(i%)+SPACE$(4091)
 12:      newheader$="Header"+STR$(i%)+SPACE$(81)
 13:      nr$=SPACE$(2)
 14:      fname$=SPACE$(32)
 15:      newfont_adr%(i%)=VARPTR(newfont$)
 16:      newheader_adr%(i%)=VARPTR(newheader$)
 17:      nr_adr%=VARPTR(nr$)
 18:      fname_adr%=VARPTR(fname$)
 19:      f$="\GEMFONTS\GEM_F"+STR$(i%+2)+".FNT"
 20:      OPEN "I",#1,f$
 21:      BGET #1,nr_adr%,2
 22:      BGET #1,fname_adr%,32
 23:      BGET #1,newfont_adr%(i%),4096
 24:      CLOSE #1
 25:      LPOKE header_adr%+84,newheader_adr%(i%)
 26:      BMOVE header_adr%,newheader_adr%(i%),88
 27:      DPOKE newheader_adr%(i%),DPEEK(nr_adr%)
 28:      BMOVE fname_adr%,newheader_adr%(i%)+4,32
 29:      LPOKE newheader_adr%(i%)+76,newfont_adr%(i%)
 30:      DPOKE newheader_adr%(i%)+66,12
 31:      header_adr%=newheader_adr%(i%)
 32:    NEXT i%
 33:  RETURN
```

Listing 2: Procedure Font

```
 1:  ' STAD:GEM.GFA
 2:  ' PROGRAMM ZUM UMWANDELN VON STAD-FONTS IN GEM-
     FONTS
 3:  ' HEINZ KATZENMAYER, 10/88
 4:  '
 5:  ' Ordner anlegen:
 6:  MKDIR "GEMFONTS"
 7:  font%=1
 8:  '
 9:  REPEAT
10:    CLS
11:    '
12:    ' Speicherplatz fuer Fontdaten:
13:    font_adr$=SPACE$(4096)
14:    help_adr$=SPACE$(4096)
15:    font_adr%=VARPTR(font_adr$)
16:    help_adr%=VARPTR(help_adr$)
17:    '
18:    ' STAD-Font auswaehlen:
19:    FILESELECT "\STADFONT\*.FNT","",fstad$
20:    IF fstad$=""
21:      END
22:    ENDIF
23:    '
24:    ' Name von Font fuer Fontheader:
25:    n$=SPACE$(32)
26:    LSETn$=MID$(fstad$,11,INSTR(fstad$,".")11)+
           ".STAD"
27:    '
28:    ' Fontnummer:
29:    INC font%
30:    '
31:    PRINT AT(20,8);"Das STAD-Font ";LEFT$(n$,INSTR(
          n$,".")-1)
32:   PRINT AT(20,10);"wird neu abgespeichert
                       im Ordner GEMFONTS"
33:    '
34:    ' Fontheader-Eintrag Nummer und Name:
35:    header$=MKI$(font%)+n$
36:    '
37:    ' STAD-Fontdaten einlesen ...
38:    OPEN "I",#1,fstad$
39:    BGET #1,help_adr%,4096
40:    '
41:    ' ... und auf GEM-Format aendern:
42:    b%=-1
43:    FOR zeichen%=0 TO 255
44:      FOR scan%=0 TO 15
45:        INC b%
46:        POKE font_adr%+zeichen%+scan%*256,
                                    PEEK(help_adr%+b%)
47:      NEXT scan%
48:    NEXT zeichen%
49:    '
50:    ' Fontdaten plus Header in GEMFONT abspeichern:
51:    fgem$="\GEMFONTS\GEM_F"+STR$(font%)+".FNT"
52:    OPEN "O",#2,fgem$
53:    BPUT #2,VARPTR(header$),34
54:    BPUT #2,font_adr%,4096
55:    CLOSE
56:    '
57:    ' Noch ein Font umwandeln?:
58:    al1$="Das STAD-Font "+LEFT$(n$,INSTR(n$,".")-1)
59:    al2$="|ist abgespeichert unter: "+"|"
60:    al$=al1$+al2$+fgem$
61:    ALERT 3,al$+"|Noch ein Font umwandeln?",1," JA |
                 NEIN",al%
62:  UNTIL al%=2
63:  CLS
64:  END
```

Listing 3: STAD: GEM. GFA

ENDE

Public-Domain-Software
»»» zu steinharten Preisen!!«««

Sie kennen uns noch nicht?
Dann nutzen Sie unser PD-Werbeangebot:

4 Atari-ST-Public-Domain-Disketten

gefüllt mit tollen Spielprogrammen, Utilities und Anwendersoftware – im Spezialverfahren kompaktiert auf einer 2-seitigen 3 1/2"-Diskette, dazu unsere ausführliche PD-Liste.
...alles zusammen bekommen Sie gegen Einsendung von
NUR DM 5,–
in Briefmarken
Scheck oder Bar!

incl. Disk, Porto und Verpackung!

PD-Liste gratis!

Viel Artware und Signumzeichensätze bei uns erhältlich!

Hotline:
Tel.: 0781/58345



FsKS Ludwig Abt. Atari, Kastanienallee 24, D-7600 Offenburg

Über 600 PD-Disks für den ATARI ST

PUBLIC DOMAIN

- Riesiges Soft- & Hardwareangebot !!
- Die besten Anwendungen und Spiele...
- PD-Software für Erwachsene..., Niedrigstpreise!

PD-Extrapakete

Nr. 1: Enthält 10 Markendisketten 1dd mit vielen tollen und neuen PD-Spielen!

Nr. 2: Enthält 10 Markendisketten 1dd mit Anwendungen, Acc's, Spielen, Utilities, heißen Girls und vieles mehr!

je Paket nur 49,90 DM
zuzüglich Porto- & Versandkosten! (siehe unten).

Hier nun weitere Angebote:

Signum II	359,90	STAD V1.2	149,90
T.I.M. V1.1	259,90	Cashflow	259,90
Banktransfer	259,90	Depot	439,90
Monitorumschalter	39,90	PAL-Interface III	189,90

Lieferung zzgl. 5,– DM bei Vorauskasse bzw. 7,– DM bei Nachnahme!

10 Top PD-Spiele oder starke Anwendungen!!!
Erhalten Sie für nur je 10,–DM (Scheck oder Schein) auf eine Markendiskette plus unseren 40-seitigen Hauptkatalog! Bitte auswählen! Für 1,60 DM in Briefmarken gibts nur unseren Hauptkatalog!

Computer-Software ☆ Ralf Markert
☆ Balbachtalstr. 71 ☆ 6970 Lauda ☆ ☎ 09343 / 3854 ☆

Public-Domain Software

- Alle Programme werden auf erstklassigem Diskettenmaterial geliefert!
- PD-Software.

Jede Einzeldiskette (SS oder DS formatiert)
nur: DM 5,99,–
Preis je 10er Block DS formatiert; 5 Disketten
nur: DM 29,99,–
Preis je 10er Block SS formatiert; 10 Disketten
nur: DM 49,99,–

Senden Sie Ihre Bestellung bitte an:

**Active Software
Ostpreußenstr. 4
8400 Regensburg**

[] Nachnahme: (Nur Inland zuzügl. DM 5,- Nachnahmegebühr).
[] Scheck über DM .... liegt bei.
- Alle Preise inklusive Porto und Verpackung.

Digital Image

EDV-Service bietet an:

- ★★ PD-Disk's nur 5,– pro Disk (alle ST-Computer PD's)
- ★★ Copy-Service von 3.5 auf 5,25 Z. nur 8,– pro Disk (IBM)
- ★★ EDV-Komplettinstallationen für kleinere Gewerbebetriebe
- ★★ Softwareproblemlösungen
- ★★ Hardwarevertrieb der Firmen Atari, Tandon, NEC, Star
- ★★ Computeranimationen für Messen. Präsentationen und Ausstellungen

**Tel. (06142) 22636 & 43560
Postfach 1206
6096 Raunheim am Main**

ATARI ST SOFTWARE

AS-HAUSHALT (Haushaltsbuchführung) — DM 99.–
Buchführung für alle Privathaushalte. Buchen über Konto/Gegenkonto. Suchen und Löschen von Buchungen. Auto-Save. Monats- und Jahresabschluß. Druckeranpassung. Journalausdruck. Balkengrafik. Deutsches Handbuch.

ST-FIRMENBUCH (Buchführung) — DM 139.–
Doppelte Buchführung für kleinere Gewerbetreibende. Automatisches Mitführen von MwSt-Konten. 5 versch. MwSt-Sätze frei einstellbar. Universelle Druckeranpassung. Abschlußzeitraum Monat/Quartal/Jahr. Umsatzsteuervoranmeldung. Journalausdruck. Ausführliches deutsches Handbuch.

ST-ÜBERWEISUNGSDRUCK — DM 45.–
Bedruckt alle Arten von Überweisungsträgern, Schecks, Zahlkarten usw. Universelle Druckeranpassung, eigene Formulare können mit einem Texteditor selbst erstellt werden. Automatische Umwandlung des Betrages in Zahlwort. Eingaben sind abspeicherbar.

DATENBANKANWENDUNGEN — DM 69.–
6 Anwendungen für Adimens ST, z.B. Termin-Adressenverwaltung, Lager.

ST-SCHREIBMASCHINE — DM 59.–
Mit diesem Programm arbeitet Ihr ST+Matrixdrucker wie eine Schreibmaschine. Sehr gut geeignet zum Ausfüllen von amtlichen Formularen. Ausdruck entweder direkt oder über editierbares Display. Mit dt. Handbuch.

ST-ETIKETT — DM 59.–
Bedruckt alle Arten von Etiketten. Jede Art von Etiketten kann separat mit einem Texteditor definiert werden. Mit Seriennummerngenerierung.

ST-AUFTRAG (Fakturierung) — DM 398.–
Eine komfortable und einfach zu handhabende Fakturierung unter GEM. Ausgabe von Listen, Serienbriefen, Etiketten sowie Fakturformularen mit 1st Word Plus Schriftattributen. Alle Formulare sind selbst anpaßbar. Demoversion mit Handbuch DM 40.- (wird angerechnet).

SOFORT KOSTENLOSE INFO 4/88 ANFORDERN

AS DATENTECHNIK
AS-DATENTECHNIK • Mainzer Str. 69
6096 Raunheim • Tel. 06142/22677

Schulmeister ST

Atari ST (Mega ST) ,500 K Ram sw - Monitor . Die Noten- und Klassenverwaltung mit Pfiff. Ein flexibles. bewährtes Konzept für Lämpels aller Schulstufen. Lassen Sie Ihren Rechner die tägliche Routinearbeit erledigen , damit Sie sich Ihren pädagogischen Aufgaben widmen können. Auch für die Schweiz geeignet !

Ausführliche Information mit Freiumschlag anfordern bei:

M. Heber-Knobloch
Auf der Stelle 27
7032 Sindelfingen



★ **Super-Lohn-Einkommensteuer** ★
Jahresausgleich 88 mit Kundenverw., Analyse, Formulardruck, jährl. Aktuelle Demo-Disk,

★ **Rentenber. – Beamtenversorgung** ★

★ **Miet-Lasten-Wohngeldberechn.** ★
Mit allen Kreisen und Gemeinden der BRD

★ **Sonderausgabenoptimierung** ★
Vorsorgeanalyse für Versicherungsnehmer.

Info gegen Rückporto Disk ab DM 80,–

**H-I SOFTWARE · Niederfelderstr. 44
8072 Manching · Telefon 08459/1669**

Softwareversand Melchart

Innsbrucker Straße 32 - 8230 Bad Reichenhall
Tel. 08651/64514 (09.30-10.30 Uhr) (14.30-18.00 Uhr)

Public Painter Monochron	74,00
CopyStar v3.0	144,00
PR's Disk-Speeder	84,00
GFA-Assembler	129,00
Tempus v2	109,00
TIM II	578,00
Campus	798,00
GFA-Raytrace	129,00

Programmiersprachen

Megamax Modula-2	379,00
Megamax Laser-C-Compiler	379,00
GFA Assembler	129,00

Kopierprogramme

CopySTar v3	144,00
G Copy	88,00

Anwenderprogramme

TiM v1	259,00
Cashflow	259,00
Banktransfer	259,00
Computer Dictionary	89,00

Utilities

Harddisk Utility v1.3	29,00
Harddisk Utility v2	59,00
GFA-Farbkonverter	52,00
GFA-Monochromkonverter	52,00
PRs Disk Speeder	84,00

Spiele

Carrier Command	69,95
Die Arche des Captain Blood	69,95
Exolon	62,95
Hostile	69,95
Kaiser	119,00
Menace	59,95
Nebulus	62,95
Starglider II	79,95
Virus	59,95

Neuerscheinungen !
Daley Thompson'S Olympic Challenge • Emmanuelle • Fugger • Iron Lord • 007 Live and let die • Operation Neptun • The Last Ninja

Weihnachtspakete
z.B. Mewilo • Sidewinder 89,95
und viele mehr! • Kostenloses Info!

Bei Vorauskasse keine Versandkosten!
Fordern Sie unsere kostenlose Preisliste an!

HSS

Hardware-Software Systemlösungen
Harmel-Scollar & Schmithals GbR
Darmstädter Str. 20 · 5000 Köln 1 · ☎ 0221/316207

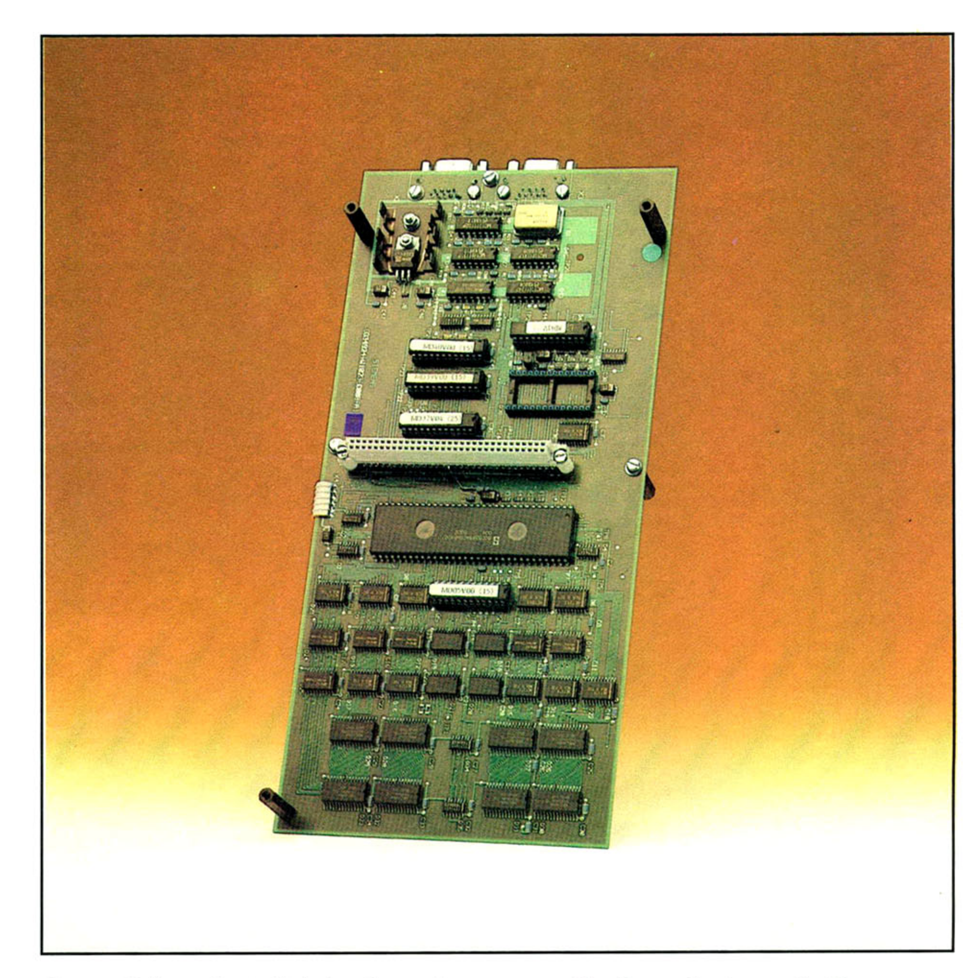
Application	Daily Mail	159,–
	Harddisk Utility	65,–
	Megamax Modula II	349,–
	Signum II	369,–
	Stad 1.3	159,–
Atari	Adimens ST 2.1	188,–
	Calamus	379,–
	1.st Word Plus & 1.st Mail	188,–
CCD	Tempus 2.0	109,–
GFA	GFA-Basic 3.0	177,–
	GFA_Draft Plus 3.0	298,–
Omikron	Basic-Compiler 3.0	169,–
Star-Division	Star-Writer ST	188,–
Steinberg	alle Software	a. A.
Technobox	Campus Draft	139,–
	Campus CAD 1.3	774,–
Vortex	HDPlus 30, 30 MB	1298,–
	Wechselplatte 44 MB	a. A.
	Streamer 60 MB	a. A.

Weitere Hardware von Atari, Star, Nec, Tandon und sämtliche Atari Software auf Anfrage.
Versandbedingungen: Preise inkl. Versandkosten (Scheck), bei Nachnahme zuzügl. DM 5,-, alle Preise für Atari ST-Version.

SUCHT!

Der Matrix-MatScreen/M110-Großbildschirm

Es war auf der ATARI-Messe in Düsseldorf. Vor den Messehallen plätscherte geruhsam der Rhein, das Wetter kündigte den Herbst an, obwohl die Sonne noch angenehm schien. Am Morgen, zu einer der Ruhe des Spätsommers angemessenen Zeit, ging ich dann hinein. Doch schon beim ersten Blick auf einen Stand erfaßte mich Unruhe. Dort stand ein in beigem Schlaglack lackierter Blechkasten mit einer schwarzglänzenden Glasfront, mit Formen, die für einen Computerbesitzer aus der Provinz das Paradies verheißen mußten.



So also begann meine erste Begegnung mit diesem obskuren Objekt der Begierde. Natürlich ging ich näher heran und tat alles um seine Aufmerksamkeit zu erringen. Schließlich erfuhr ich, daß es sich um den 19"-Großbildschirm aus der Matrix-Familie handelte, der die wahrhaft umwerfende Auflösung von 1280*960 Punkten besitzt und auch noch eine große Schwester mit 24" sein eigen nennt.

Nun, nach einem kurzen aber heftigen Flirt gelang es mir, den Bildschirm zu einer kurzen Wochenendreise auf mein Landgut einzuladen. Jetzt steht er vor mir auf meinem Schreibtisch, und wir sind sehr glücklich miteinander, auch wenn unsere Beziehung nicht immer problemlos ist.

Zuerst einmal mußten Kontakte geknüpft werden. Dies gestaltete sich glücklicherweise nicht schwierig, man benötigte allerdings einen Mega-ST, möglichst mit abgelaufener Garantie, denn das Gerät muß geöffnet werden, um die Matrix-Interface-Karte in den ST-internen Erweiterungs-Slot zu stecken. Das war eigentlich schon alles. Allerdings ist in der Beschreibung des Matrix-Schirms eine unerfreuliche Möglichkeit erwähnt, die Störungen des Betriebes in Zusammenhang mit dem Laserdrucker ergibt. In diesem Fall muß nämlich im Controller des Laserdruckers ein wenig gelötet werden. Aber SLM804-Besitzern ist ja nicht neu, daß ihr Liebling häufig Ärger macht...

Zu dem Großbildschirm gehört eine Boot-Diskette, die ein verändertes TOS und ein Startup-Programm enthält. Das Startup-Programm erlaubt den Betrieb in mehreren Modi: Man kann zum einen den M110 alleine betreiben, so daß der SM124 auf dem Matrix emuliert wird. Das sieht ausgesprochen lustig aus, als ob der ganze Desktop unter einer Lupe läge. In diesem Modus läuft so ziemlich alles. Man kann den M110 auch alleine in seiner vollen Pracht verwenden. Dann laufen nur sauber programmierte Anwendungen. Es gibt auch eine große Gruppe von Programmen, die zwar laufen, aber den Bildschirm nicht ausnutzen können, weil der Programmierer, statt brav festzustellen, welche Auflösung die aktuelle VDI-Workstation besitzt, einfach mit festen Screen-Koordinaten gerechnet hat. Schließlich kann man auch den SM124 alleine verwenden - alles wie gehabt. Der letzte Modus benutzt beide Bildschirme, den SM124 wie gewohnt. Der Großbildschirm kann jetzt in einem entsprechend programmierten Programm als zweite Ausgabestation verwendet werden. Mir ist allerdings keine Software bekannt, die den M110 in diesem Modus unterstützt.

Eine gute Idee ist der Maustreiber: Wenn die Mausgeschwindigkeit auf einem

Großbildschirm konstant ist, wird es immer recht mühsam, zur Bildschirmmitte und zurück zu fahren. Deswegen bewegt sich die Maus auch überproportional schneller, wenn man heftige Mausbewegungen macht.

Ein Mangel ist die Anleitung: Sie verrät zum Beispiel nichts darüber, wie man den Matrix-Schirm im zuletzt genannten Modus programmiert. Ein paar erläuternde Worte darüber, was saubere Programmierung im Sinne der Matrix-Kompatibilität bedeutet, wären durchaus angebracht.

Es begab sich zu dieser Zeit, daß mir, ein ungeahnter Glücksfall in dieser Konstellation, ein Programm namens Calamus in die Hände fiel, das eines Testes harrte. Glücklicherweise erwies es sich als sehr offen gegenüber meinem Matrix-Schirm, so daß wir schöne Stunden vollbrachten. Einige Zeit mußte ich mit Calamus ohne den Matrix-Bildschirm zurechtkommen, die Entzugserscheinungen ließen mich schnell zu ihm zurückkehren.

Auch mein Turbo-C-Compiler erwies sich als freudiger Partner. Mehr als eine Druckseite Source-Code konnten wir gleichzeitig gemeinsam betrachten, eine Arbeitsweise, die das Programmieren zu einer wahren Lust machte. Die Freude wurde durch gelegentliche Abstürze ein wenig getrübt, aber die schöne Zeit überwog doch deutlich.

Ähnlich war es mit unser aller Freund, dem guten alten Wordplus. Leider gerietem ihm in der ungewohnten Umgebung die Seitennummern hoffnungslos durcheinander, auch ließ sein Gesundheitszustand es geraten erscheinen, wieder in die vertraute Umgebung zurückzukehren.

Leider müssen heutzutage noch viele andere Programme dieses Leiden teilen. So viele sind erzogen im Geiste des Sektierertums, der Abgrenzung und Intoleranz gegen das Ungewöhnliche. Doch diejenigen, die es gelernt haben, sich anzupassen, belohnen mit einem besonderen Genuß. DTP ist nicht mehr vorstellbar ohne den Großbildschirm. Bittere Tränen und schweres Leid mußte ich erdulden, als mein geliebter Matrix-Schirm wieder von mir scheiden mußte.



Der Matrix-Großbildschirm ist leider zu teuer, um ihn einfach der Freude wegen zu kaufen. Wer aber eine Anwendung (mit entsprechend angepaßter Software!) besitzt, die schon des Konzeptes wegen nach einem großen Monitor schreit (DTP, Grafik usw.), sollte sich unbedingt mit dem Gedanken vertraut machen, daß solche Arbeiten erst auf einem Großbildschirm wirklich komfortabel und professionell ausführbar ist. Durch den Slot-Einbau ist auch die Hardwarelösung robust und unauffällig. In Zukunft wird es sicherlich mehr Anwendungen geben, die den Bildschirm direkt unterstützen. Vielleicht kommt ja auch einmal eine Zeit, zu der auch 19"-Schirme in die Rubrik 'Power without the price' fallen.

CS

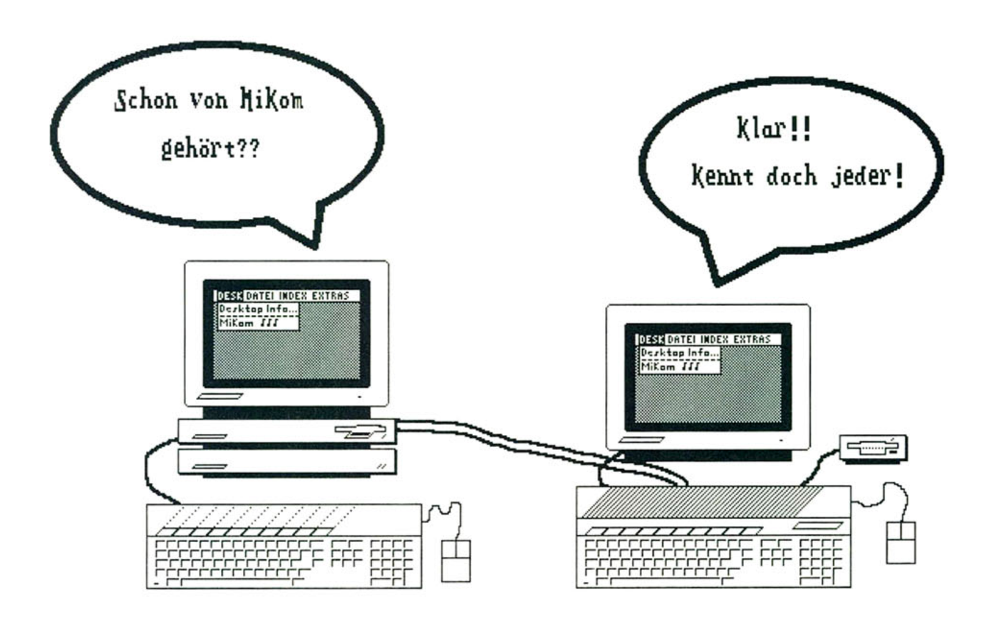
Bezugsadresse:

Matrix Datensystem GmbH
Aichelbachstr. 2
7155 Oppenweiler
Tel.: 07191/4088

ENDE

Doppelt gemoppelt hält besser

oder wie man zwei Computer miteinander verbindet



Jetzt, wo die Megas so richtig in die heimischen Computerstuben Einzug gehalten haben, finden sich sicherlich bei vielen ST-Freunden zwei STs.

Aber nicht nur Besitzer zweier STs sollen angesprochen werden, sondern auch alle diejenigen, die mal zwei Rechner miteinander verbinden möchten.

Das hier vorgestellte Programm benutzt den MIDI-Ein- und Ausgang. Außerdem ist das Programm erweiterungsfähig, es lassen sich also gegebenenfalls noch viele Routinen implementieren.

Zu den Hardware-Voraussetzungen zählen - wie zu erwarten - zwei ATARI STs und zwei Stereo-Verbindungskabel. Das Programm wurde mit Megamax Modula 2 geschrieben und mit einem Mega ST 2 (mit TOS vom 22.4.'87) und einem 520 ST+ (mit TOS vom 6.2.'86) getestet, wobei sich dort keine großen Schwierigkeiten zeigten.

Das Programm - oder besser Accessory - "MiKom" hat nur eine Funktion, und die ist auch noch vergleichsweise bescheiden. Es kann eine kleine Nachricht von einem ST zum anderen und umgekehrt übermittelt werden.

Bevor wir überhaupt etwas ausprobieren können, muß das Programm abgetippt und übersetzt sein. Das Accessory und das dazugehörige Resourcefile werden auf die Bootdisketten der beiden STs kopiert. Die beiden Rechner werden abgeschaltet und jeweils MIDI-Out des einen mit MIDI-In des anderen über je ein Stereo-Übertragungskabel verbunden. Nun die Rechner wieder einschalten und mit den neuen Bootdisketten starten. Zu diesem Zeitpunkt stehen die beiden Rechner kommunikationsbereit vor uns (s.o.).

Um die Kommunikation aufzunehmen, wird der Accessory-Eintrag "MiKom" bei einem der beiden STs angeklickt (Bild 2), worauf eine Dialogbox erscheint. Hier kann man auswählen, was man machen möchte. Neben dem lebenswichtigen "Quit" kann eine Nachricht gesendet oder empfangen werden. Zuerst senden wir natürlich. Als nächstes erscheint eine zweite Dialogbox mit einer edierbaren Zeile und den Buttons "Quit" und Nachricht senden. Da wir senden wollen, geben

wir zuerst eine passende Nachricht ein und klicken dann "Nachricht senden" an. Auf unserem Bildschirm verschwindet die Dialogbox, und der Monitor sieht aus, als wäre nichts geschehen.

Entweder passiert nichts, und nach fünf Sekunden erscheint die Warnmeldung "Timeout", oder auf dem anderen Rechner wird eine Alertbox dargestellt, die darüber informiert, daß eine Nachricht angekommen ist. Möchte man sie sogleich betrachten, so muß "ja" angeklickt werden, sonst "nein". Für den Fall, daß "ja" gewählt wurde, erscheint die bekannte Sende-Dialogbox mit der Nachrichtenzeile des anderen ST!!! Natürlich kann sofort eine Nachricht zurückgeschickt werden, sonst sollte "Quit" gewählt werden. Im Fall "nein" verschwindet die Alertbox einfach wieder. Die Nachricht kann man sich jederzeit ansehen, wenn man den Accessory-Eintrag auswählt und "Nachricht senden/empfangen" anklickt.

Zur Bedienung war das schon alles. Vor der Zerstörung des Bildschirmaufbaus braucht man keine Angst zu haben, denn in der Dialog-Routine ist eine eigenständige Restauration vorgesehen.

Alertbox *Arrive*

♪♪♪ MIDI-KONTAKT ♪♪♪
============================
Eine Nachricht ist angekommen!
Möchten Sie sie jetzt sehen?

ja | nein

Bild 3: Daten sind auf dem Rechner empfangen worden

Nun zu dem, was für den Programmierer interessant ist. Das Modul "*DialogHandler*" (Listing 2a und 2b) soll nicht ausführlich erklärt werden, denn "*DoDialog*" stammt ursprünglich aus einem anderen Listing, welches zum Zeitpunkt der Veröffentlichung dieses Programms schon erschienen sein sollte. "*DoDialog*" wurde lediglich in ein Extra-Modul ausgelagert und minimal verändert.

Die prinzipielle Funktionsweise soll jedoch trotzdem kurz erläutert werden. Normalerweise wird für eine Dialogbox ein Bildschirmbereich reserviert, die Dialogbox ausgegeben, der Dialog durchgeführt und anschließend der reservierte Bildschirmbereich wieder freigegeben. Dazu dienen die AES-Routinen FORM_DIAL, FORM_DO und OBJC_DRAW. FORM_DIAL ist dazu da, den Bildschirmbereich zu reservieren und wieder freizugeben. Bei Freigabe wird jedoch an das Programm, dessen Bildschirmbereich zerstört wurde, eine Meldung geschickt. Eine selbständige Restauration findet nicht statt.

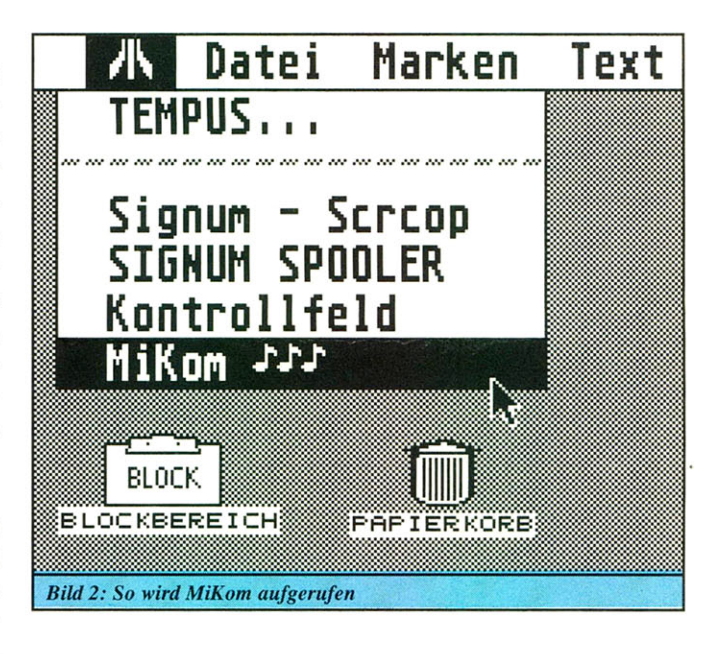


Bild 2: So wird MiKom aufgerufen

"DoDialog" führt den vollständigen Dialog durch, rettet jedoch den Bildschirmausschnitt vorher und übernimmt auch die Restauration. Wofür nun der ganze Aufwand? Es gibt ein paar Programme, die keine Accessories zulassen, aber trotzdem Events verwenden. Auch bei solchen Programmen kann eine Meldung von MiKom erscheinen. Da MiKom Dialogboxen benutzt, könnte es passieren, daß graue Felder auf dem Bildschirm bleiben. Da jedoch *"DoDialog"* den Bildschirm wiederherstellt, werden die grauen Felder vermieden.

Die beiden anderen Routinen setzen bzw. lesen einen String aus einer Object-Struktur. Sie werden benötigt, um den Text der Nachrichtenzeile zu setzen und zu ermitteln.

Viel wichtiger ist der *"MIDIHandler"* (Listing 3a und 3b) und *"MiKom"* (Listing 1a, 1b und 1c).

Das Grundprinzip ist in den beiden C-Listings dargestellt (Listing 0a und 0b). Ein Rechner sendet 1-Byte-Zeichen zu dem anderen, der diese weiterverarbeitet. Die beiden Programme arbeiten natürlich noch nicht ganz so, wie man es sich wünschen würde, denn jede andere Arbeit wird dabei unterbrochen.

Die im *MIDIHandler* implementierten Routinen helfen schon eher weiter. *WriteStringToMIDI* und *ReadStringFromMIDI* senden bzw. empfangen Zeichenketten über die MIDI-Schnittstelle, die wir im folgenden kurz MIDI nennen. Analog arbeiten *WriteToMIDI* und *ReadFromMIDI*, nur diesmal für einen einzelnen Character. *MIDIReceived* gibt TRUE zurück, falls mindestens ein Zeichen im MIDI-Empfangspuffer ist, ansonsten FALSE.

Schließlich die Routinen *SetTimeOut* und *GetTimeOut*, die die TimeOut-Zeit setzen beziehungsweise abfragen. Die Zeit wird in Sekunden angegeben. Eine Timeout-Warnmeldung wird gegeben, falls ein Zeichen gesendet oder empfangen wer-

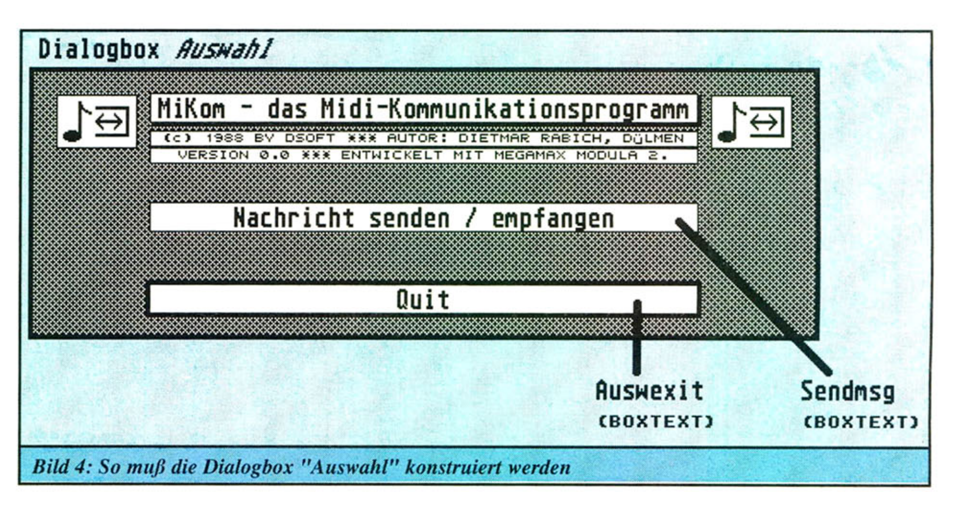


Bild 4: So muß die Dialogbox "Auswahl" konstruiert werden

den soll, aber innerhalb der Timeout-Zeit eben nicht gesendet oder empfangen werden kann. Die umfangreichsten Routinen aus diesem Modul sind *WriteStringToMIDI* und *ReadStringFromMIDI*. Bevor wir jedoch zu deren Erklärung kommen, noch etwas zu den benutzten BIOS-Routinen.

```
 1:  /** MIDI-Empfangsprogramm **/
 2:  /* Entwickelt mit Turbo C. */
 3:
 4:  # include <tos.h>
 5:  # include <ext.h>
 6:
 7:  main()
 8:
 9:  {
10:   char c;
11:
12:   for (;;)
13:    {
14:     c=Bconin(3);
15:     putch(c);
16:    }
17:  }
```

Listing 0a

```
 1:  /**** MIDI-Sendeprogramm ****/

 2:  /* Entwickelt mit Turbo C.   */

 3:
 4:  # include <tos.h>
 5:  # include <ext.h>
 6:
 7:  main()
 8:
 9:  {
10:   char c;
11:
12:   for (;;)
13:    {
14:     c=getch();
15:     putch(c);
16:     Bconout(3,c);
17:    }
18:  }
```

Listing 0b

Die BIOS-Routinen teilen sich in Status- und I/O-Routinen auf. Status-Routinen sind *Bconstat* (Bios 1) und *Bcostat* (Bios 8). *Bconstat* stellt fest, ob ein Zeichen verfügbar ist. Für den MIDI-Port muß die Gerätenummer 3 als einziger Parameter übergeben werden. Falls ein Zeichen verfügbar ist, so erhalten wir eine -1 (TRUE) zurück, sonst eine 0 (FALSE). Etwas eigenartiger ist *Bcostat*. Hier wird zwar auch der MIDI-Port abgefragt, aber bei den Probeläufen mit MiKom zeigte diese Routine keinerlei nennenswerte Wirkung. Sie ist aber trotzdem in dem Modul *MIDIHandler* deklariert. Normalerweise sollte *Bcostat* auch die Gerätenummer 3 für MIDI übergeben werden können, aber die Gerätenummer soll mit der Nummer 4 - Tastatur-Prozessor IKBD - vertauscht sein. Naja!

```
 1:  (*******************************************************************)
 2:  (* MiKom - das MIDI-Kommunikationspr.für zwei Atari ST Version 0.0 *)
 3:  (* ----------------------------------------------------------------*)
 4:  (* Autor: Dietmar Rabich, Dövelingsweg 2, D-4408 Dülmen 27/09/88   *)
 5:  (* =============================================================== *)
 6:  (* Entwickelt mit Megamax Modula 2 von Application Systems
        ///Heidelberg. *)
 7:  (*******************************************************************)
 8:  (* H a r d w a r e - V o r a u s s e t z u n g e n :              *)
 9:  (* 2 ATARI ST und zwei Stereo-Verbindungskabel (DIN-Stecker)       *)
10:  (*******************************************************************)
11:
12:  MODULE MiKom;
13:
14:  (*****************************************************)
15:  (* * * *      I   M   P   O   R   T   E     * * * * *)
16:  (*****************************************************)
17:
18:  (* Resourcefile *)
19:                      IMPORT Mikomd;
20:
21:  (* AES-Routinen *)
22:  FROM AESEvents      IMPORT MultiEvent,MessageBuffer,accOpen,
                                EventSet,Event,
23:                             RectEnterMode,TimerEvent;
24:  FROM AESForms       IMPORT FormAlert;
25:  FROM AESMenus       IMPORT RegisterAcc;
26:  FROM AESResources   IMPORT LoadResource,ResourceAddr,ResourcePart;
27:  FROM DialogHandler  IMPORT setTextString,getTextString,DoDialog;
28:
29:  (* allgemeine GEM-Routinen *)
30:  FROM GEMEnv         IMPORT RC,InitGem,DeviceHandle,GemError;
31:  FROM GEMGlobals     IMPORT PtrObjTree,PtrMaxStr,MButtonSet,GemChar,
                                SpecialKeySet;
32:
33:  (* Graphik *)
34:  FROM GrafBase       IMPORT Point,Rect;
35:
36:  (* MIDI-Routinen *)
37:  FROM MIDIHandler    IMPORT WriteStringToMIDI,ReadString FromMIDI,
38:                             WriteToMIDI,ReadFromMIDI,MIDIReceived;
39:
40:  (* sonstige Routinen *)
41:  FROM SYSTEM         IMPORT ADR;
42:  FROM Strings        IMPORT String;
43:  FROM FastStrings    IMPORT Length;
44:
45:
46:  (*****************************************************)
47:  (* * * *     V   A   R   I   A   B   L   E   N     * * * * *)
48:  (*****************************************************)
49:
50:  VAR GEMDevice       : DeviceHandle;  (* Gerätekennung  *)
51:      ReturnButton,                    (* für Dialog     *)
52:      VoidC,                           (* diverse Zwecke *)
53:      AccID           : CARDINAL; (* Accessory-Kennung   *)
54:      DialogBox,
55:      AuswahlBox      : PtrObjTree;    (* für Dialog     *)
56:      MessageArrive   : PtrMaxStr;
57:      InitOK          : BOOLEAN;  (* für Initialisierung *)
58:      AccName         : ARRAY [0..15] OF CHAR;
                                     (* Accessory-Name      *)
59:      TypeOfImport    : CHAR;     (* Art der MIDI-Daten  *)
60:
61:
62:  (*****************************************************)
63:  (* * * *      R   O   U   T   I   N   E   N     * * * * *)
64:  (*****************************************************)
65:
66:  (****** sendet Nachricht an den anderen Computer *****)
67:  PROCEDURE SendeNachricht;
68:
69:   VAR InfoString : String;
70:
71:   BEGIN
72:    ReturnButton:=DoDialog(GEMDevice,DialogBox);(*Dialog durchführen*)
73:
74:     (* Nachricht senden... *)
75:    IF ReturnButton=Okbutton THEN
76:     getTextString(DialogBox,Infoline,InfoString);
77:     IF InfoString[0]#0C THEN
78:      WriteToMIDI(1C);              (* Code senden      *)
79:      WriteStringToMIDI(InfoString)  (* String senden   *)
80:     END
81:    END
82:   END SendeNachricht;
```

Listing geht weiter...

Die IO-Routinen sind *Bconin* (Bios 2) für die Eingabe und *Bconout* (BIOS 3) für die Ausgabe via MIDI. *Bconin* wird die bekannte Gerätenummer 3 für MIDI als Parameter übergeben, dafür erhält man einen Long-Wert(!!!) zurück. Ein Long-Wert besteht bekanntlich aus 4 Bytes, und daher sollte man sich fragen, woher die anderen 3 Bytes kommen, wenn wir nur einen 1-Byte-Wert senden. Die Bits 16 bis 23 werden für den Scancode genutzt, falls das Eingabegerät *Konsole* Verwendung findet, und die Bits 24 bis 31 liefern den Wert der Bios-Routine *Kbshift*, falls das passende Bit in der Systemvariablen *conterm* gesetzt ist. Wofür die Bits 8 bis 15 da sind, ist etwas unklar, aber wir benötigen sowieso nur die Bits 0 bis 7, die unser Zeichen enthalten. Die Ausgaberoutine *Bconout* ist einfacher aufgebaut, denn ihr wird neben der MIDI-Gerätenummer nur der Wert unseres Characters als Parameter übergeben. Mit diesen vier Routinen wird all das, was wir so brauchen, bewerkstelligt.

Zum Senden und Empfangen sind die Routinen *WriteToMidi* und *ReadFromMidi* da, die jedoch nur intern benutzt werden. *WriteToMidi* versucht, ein Zeichen über MIDI zu schreiben. Dazu wird in einer Schleife abgewartet, ob der Puffer bereit ist. Ist er es innerhalb der Timeout-Zeit nicht, so wird eine Warnmeldung ausgegeben und kein Zeichen gesendet. Ist der Puffer bereit, so wird das Zeichen abgeschickt. *ReadFromMidi* arbeitet analog, nur empfängt diese Routine, anstatt zu senden.

WriteStringToMIDI sendet einen String über MIDI. Damit der andere Rechner weiß, wieviele Bytes geschickt werden, wird zuerst die Länge übermittelt und danach der String, dieser natürlich auch wieder zeichenweise. Sollte sich mal ein Zeichen nicht senden lassen, bricht die Routine ihre Arbeit einfach ab. Ist alles in Ordnung, wird auf eine Bestätigung vom anderen Rechner gewartet, der ein Zeichen zurücksendet, wenn alles korrekt übermittelt wurde. Analog dazu arbeitet wieder *ReadStringFromMIDI*. Hier wird eine Zeichenkette gelesen und im Anschluß daran das Bestätigungszeichen geschickt.

WriteToMIDI und *ReadFromMIDI* arbeiten wie die entsprechenden String-Routinen, nur eben für einen einzelnen Character.

Damit ist das Geheimnis des *MIDIHandlers* gelüftet. Kommen wir nun zu dem eigentlichen Accessory: *MiKom*.

```
 83:
 84:  (*** empfängt Nachricht von dem anderen Computer   ***)
 85:  PROCEDURE EmpfangeNachricht;
 86:
 87:   VAR InfoString : String;
 88:
 89:   BEGIN
 90:    ReadStringFromMIDI(InfoString);    (* Daten holen     *)
 91:    setTextString(DialogBox,Infoline,InfoString);
                          (* String in Dialogbox setzen  *)
 92:    FormAlert(1,MessageArrive^,VoidC);
 93:    IF VoidC=1 THEN      (* Nachricht ansehen?         *)
 94:     SendeNachricht      (* ggf. Nachr. zurücksenden    *)
 95:    END
 96:   END EmpfangeNachricht;
 97:
 98:  (**************** Hauptroutine *********************)
 99:  PROCEDURE Ablauf;
100:
101:   VAR Message      : MessageBuffer;
                                    (* Puffer für Message *)
102:       Events       : EventSet;  (* Set für Events     *)
103:       dummyPoint   : Point;     (* Dummy-Variable     *)
104:       dummyBut     : MButtonSet; (*     "              *)
105:       dummyChar    : GemChar;   (*     "              *)
106:       specialKeys  : SpecialKeySet; (*  "              *)
107:
108:   BEGIN
109:    REPEAT
110:     MultiEvent (EventSet{message,timer}, (* Event abwarten  *)
111:                 0,MButtonSet{},MButtonSet{},
112:                 lookForEntry,Rect(0,0,0,0),
113:                 lookForEntry,Rect(0,0,0,0),
114:                 Message,10L,
115:                 dummyPoint,dummyBut,specialKeys,dummyChar,
116:                 VoidC,Events);
117:
118:     (* Falls unser Accessory angewählt wurde...       *)
119:     IF (message IN Events) AND (Message.msgType=accOpen) THEN
120:
121:      (* Auswahldialogbox darstellen *)
122:      ReturnButton:=DoDialog(GEMDevice,AuswahlBox);
123:
124:      (* Auswertung *)
125:      CASE ReturnButton OF
126:       Sendmsg : SendeNachricht|
127:
128:      (* Hier können weitere Aktionen eingefügt werden. *)
129:
130:      END
131:
132:     END;
133:
134:     (* Falls vom anderen ST etwas via Midi kam...      *)
135:     IF (timer IN Events) AND MIDIReceived() THEN
136:
137:      (* Code ermitteln *)
138:      ReadFromMIDI(TypeOfImport);
139:
140:      (* Auswertung *)
141:      CASE TypeOfImport OF
142:
143:       (* Es ist eine Nachricht angekommen...           *)
144:       1C : EmpfangeNachricht|
145:
146:       (* Auch hier ist noch Platz für weitere Aktionen *)
147:       (* Das erste Byte bestimmt,  was gesendet  wurde *)
          (* und was damit bewirkt werden soll.           *)
148:       (*   "1"  heiPt somit: Nachricht lesen und       *)
149:       (* ausgeben.                                    *)
150:       (* Beispielsweise könnte man mit "2" ganze
             Dateien übertragen.                          *)
151:       (* Dazu ist reicht es, hier eine weitere Aktion an
             zufügen.Die Auswahldialogbox sollte dann auch *)
152:       (* um die weiteren Möglich-                     *)
153:       (* keiten erweitert werden.                     *)
154:
155:      END
156:
157:     END
158:
159:    UNTIL FALSE
160:   END Ablauf;
161:
162:
```

```
163:  (********************************************************)
164:  (* * * * *    H a u p t p r o g r a m m      * * * * *)
165:  (********************************************************)
166:
167:  BEGIN
168:   InitGem(RC,GEMDevice,InitOK);(* anmelden *)
169:   IF InitOK THEN
170:    LoadResource ('MIKOM.RSC'); (* Resourcefile laden *)
171:    IF GemError() THEN             (* OK? *)
172:     FormAlert(1,'[0][Das Resourcefile fehlt!][ OK ]',VoidC);
173:     REPEAT
174:      TimerEvent(10000L)
175:     UNTIL FALSE
176:    ELSE
177:     AccName:='  MiKom ♪♪♪ ';
                                (* Accessory-Name setzen              *)
178:     RegisterAcc(ADR(AccName),AccID,InitOK);(* Accessoryanmelden *)
179:     IF InitOK THEN             (* OK?                              *)
180:      DialogBox :=ResourceAddr(treeRsrc,Dialog);
                                (* Dialogbox-Adresse                  *)
181:      AuswahlBox:=ResourceAddr(treeRsrc,Auswahl);
182:      setTextString(DialogBox,Infoline,'');
                                (* Leerstring setzen                  *)
183:      MessageArrive:=ResourceAddr(textString,Arrive);
                                (* Alertbox-Adresse                   *)
184:      Ablauf                    (* Accessory-Ablauf                   *)
185:     ELSE
186:      FormAlert(1,'[0][Accessory MiKom|nicht installiert.]
                                               [ OK ]',VoidC);
187:      REPEAT
188:       TimerEvent(10000L)
189:      UNTIL FALSE
190:     END
191:    END
192:   END
193:  END MiKom.
194:
195:  (* * * * *    E n d e   d e s   M o d u l s   * * * * *)
196:
```

Listing 1a

```
 1:  DEFINITION MODULE Mikomd;
 2:
 3:  EXPORT
 4:    Dialog, Infoline, Exbutton, Okbutton, Auswahl, Sendmsg,
 5:    Auswexit, Arrive;
 6:
 7:
 8:  CONST
 9:    Dialog   =   0;  (* Formular/Dialog *)
10:    Infoline =   7;  (* FTEXT in Baum DIALOG *)
11:    Exbutton =   8;  (* BUTTON in Baum DIALOG *)
12:    Okbutton =   9;  (* BUTTON in Baum DIALOG *)
13:    Auswahl  =   1;  (* Formular/Dialog *)
14:    Sendmsg  =   8;  (* BOXTEXT in Baum AUSWAHL *)
15:    Auswexit =   9;  (* BOXTEXT in Baum AUSWAHL *)
16:    Arrive   =   0;  (* Meldung-String Index *)
17:
18:  END Mikomd.
19:
20:  (* Listing 1b *)
```

Listing 1b

```
1:  IMPLEMENTATION MODULE Mikomd;
2:  (*$N+,M-*)
3:  END Mikomd.
4:
5:  (* Listing 1c *)
```

Listing 1c

Bevor das Listing zu *MiKom* angefertigt wird, sollte ein Resourcefile erstellt werden. Dieses Resourcefile muß zwei Dialogboxen und eine Alertbox enthalten. Ein Muster ist auf den Bildern 3 bis 5 zusehen. Das Resource-Construction-Programm legt auch die beiden Module an, die als Listing 1b und 1c aufgeführt sind. Das Resourcefile wurde übrigens unter dem Namen *Mikomd* angelegt, um eine eventuelle Verwirrung zu vermeiden. Es ist vor der Benutzung entsprechend umzubenennen.

Das Modul *MiKom* umfaßt neben dem Hauptprogramm lediglich drei Routinen. *SendeNachricht* sendet eine Nachricht. Damit aber *MiKom* erweiterungsfähig bleibt, wird vor der Nachricht eine Kennung übermittelt. Sie besteht hier aus einer 1. Wird sie geschickt, weiß der andere Rechner gleich Bescheid, daß eine Nachricht kommt und nicht etwa eine Datei oder sonst etwas. *EmpfangeNachricht*, die zweite Routine, arbeitet ähnlich, nur taucht hier die Kennung nicht auf.

Die gesuchte Kennung steckt in der Routine *Ablauf*. Sie enthält eine Endlosschleife, in der auf einen Message- bzw. Timer-Event gewartet wird. Tritt also ein Timer-Event ein, wird abgefragt, ob ein Zeichen am MIDI-Port anliegt. Falls ja, wird es gelesen. Dieses Zeichen ist die Kennung, die der andere Rechner geschickt hat! Also kann an dieser Stelle entschieden werden, wie weiterverfahren werden soll. Da *MiKom* aber in der aktuellen Fassung nur eine Nachricht empfangen kann, wird auch nichts anderes als Alternative aufgeführt. Die auf den ersten Blick unsinnige CASE-Anweisung hat einen besonderen Sinn. Sollen weitere Möglichkeiten angefügt werden, können diese durch CASE einfacher eingefügt werden. Falls ein Message-Event eingetreten ist, kann auch eine bestimmte Routine ausgewählt werden. Aber auch hier ist bisher nur das Senden einer Nachricht möglich. CASE ist aus den gleichen Gründen, die gerade schon angeführt wurden, benutzt worden.

An den beiden genannten Stellen lassen sich also leicht weitere Routinen einfügen. Es ist dann aber unbedingt auch die Auswahl-Dialogbox entsprechend zu erweitern. Auf Erweiterungen kommen wir aber gleich noch wieder zurück.

Das Hauptprogramm hat lediglich die ehrenvolle Aufgabe, das Accessory zu installieren und entsprechend vorzubereiten. Zu den Vorbereitungen gehört das Laden des Resourcefiles und das Festle-

gen der Adressen, aber auch das Anmelden des Accessorys.

Nun sind wir am Ende der Erklärungen angelangt. Was kann man an Erweiterungen einbringen? Tja, eine gute Frage. Hier können nur einige wenige Vorschläge gemacht werden. Denkbar wäre beispielsweise das Übermitteln von Dateien. Dabei könnte die Möglichkeit berücksichtigt werden, daß *MiKom* selbständig das Inhaltsverzeichnis einer Diskette, die im Laufwerk des anderen ST liegt, ausliest. Es könnten aber auch direkt Hardcopies übertragen werden. Möchte man mehr als zwei STs benutzen, muß ein etwas größerer Aufwand betrieben werden. Neben der Kennung dessen, was übertragen wird, müßte ein Byte eingeführt werden, das den Adressaten bestimmt. Gleichzeitig sollte man ein Freizeichen senden, beispielsweise eine 0. Lediglich der ST, der gerade die 0 hat, kann etwas absenden. Damit hätte man dann schon so ein richtiges kleines Netz. Nachteilig ist natürlich, daß immer alle Rechner angeschaltet sein müssen und dabei nur GEM-Programme verwenden dürfen. Bei TOS-Programmen, die keine Events nutzen, steht ja auch *MiKom* still.

Die MIDI-Schnittstelle kann noch auf vielerlei Art und Weise genutzt werden. Ich hoffe, daß dieser Artikel Anreiz genug bietet, viele eigene Versuche zu starten.

Dietmar Rabich

Literatur:

[1] *Atari ST Profibuch*
H.-D. Jankowski/ J.F. Reschke/ D. Rabich
Sybex, 5. Auflage 1988

[2] *Auf der Schwelle zum Licht - Zeichenorientierte Geräte*
A. Esser
ST Computer 10/88

Bild 5: So muß die Dialogbox "Dialog" konstruiert werden

```
 1: (*************************************************************** )
 2: (*Definitionsmodul für Dialog-Routinen Version 0.0,19/09/88  *)
 4: (*============================================================ *)
 5: (*Autor: Dietmar Rabich, Dülmen.Entw. mit Megamax Modula-2   *)
 7: (**************************************************************)
 8: (*Erläuterungen befinden sic im Implementationsmodul          *)
 9: (**************************************************************)
10:
11:  DEFINITION MODULE DialogHandler;
12:
13:  FROM GEMEnv      IMPORT DeviceHandle;
14:  FROM GEMGlobals  IMPORT PtrObjTree;
15:
16:  PROCEDURE getTextString (tree    : PtrObjTree;
17:                           obj     : CARDINAL;
18:                           VAR str : ARRAY OF CHAR);
19:
20:  PROCEDURE setTextString (tree : PtrObjTree;
21:                           obj  : CARDINAL;
22:                           str  : ARRAY OF CHAR);
23:
24:  PROCEDURE DoDialog (GEMDevice : DeviceHandle;
                           tree :PtrObjTree):CARDINAL;
25:
26:  END DialogHandler.
27:
28:  (* Listing 2a *)
```

Listing 2a

```
 1: (********************************************************************)
 2: (* Implementationsmodul für Dialog-Routinen Version 0.0, 19/09/88    *)
 3: (* ================================================================*)
 4: (* Autor: Dietmar Rabich, Dülmen. Entw. mit Megamax Modula 2.       *)
 5: (********************************************************************)
 6: (* setTextString und getTextString sind Routinen aus MShell nachempf.*)
 7: (* DoDialog stammt von D. Rabich und wurde in der ST Computer veröf. *)
 8: (********************************************************************)
 9:
10:  IMPLEMENTATION MODULE DialogHandler;
11:
12:  (* AES-Routinen *)
13:  FROM AESForms      IMPORT FormDo,FormDial,FormCenter,FormDialMode;
14:  FROM AESGraphics   IMPORT MouseForm,GrafMouse;
15:  FROM AESObjects    IMPORT ChangeObjState,DrawObject;ObjectOffset;
16:  FROM AESWindows    IMPORT UpdateWindow;
17:  FROM ObjHandler    IMPORT ObjectState,ObjectSpace,ObjTreeError,
18:                            GetTextStrings,SetCurrObjTree,
                             AssignTextStrings,
19:                            SetPtrChoice;
20:
21:  (* allgemeine GEM-Routinen *)
22:  FROM GEMEnv        IMPORT DeviceHandle;
23:  FROM GEMGlobals    IMPORT PtrObjTree,ObjState,OStateSet,Root,MaxDepth;
24:
25:  (* Graphik *)
26:  FROM GrafBase      IMPORT Rectangle,Rect,TransRect,Pnt,MemFormDef,
27:                            GetLogMemForm,BitOperation;
28:
```

```
29:  (* VDI-Routinen *)
30:  FROM VDIRasters    IMPORT CopyOpaque;
31:
32:  (* sonstige Routinen *)
33:  FROM Strings       IMPORT String;
34:  FROM Storage       IMPORT ALLOCATE,DEALLOCATE;
35:  FROM SYSTEM        IMPORT ADR,ADDRESS;
36:
37:  (* Zeichenkette in tree ermitteln, die bei obj
        steht*)
38:  PROCEDURE getTextString (tree    : PtrObjTree;
39:                           obj     : CARDINAL;
40:                           VAR str : ARRAY OF CHAR);
41:
42:   VAR voidStr : String;
43:
44:   BEGIN
45:    SetCurrObjTree(tree,FALSE);
46:    GetTextStrings(obj,str,voidStr,voidStr);
47:    IF ObjTreeError() THEN
48:     str[0]:=0C
49:    END
50:   END getTextString;
51:
52:  (* Zeichenkette in tree bei obj setzen *)
53:  PROCEDURE setTextString (tree     : PtrObjTree;
54:                           obj      : CARDINAL;
55:                           str      : ARRAY OF CHAR);
56:
57:   BEGIN
58:    SetCurrObjTree(tree,FALSE);
59:    AssignTextStrings(obj,setOnly,str,noChange,'',
                                          noChange,'')
60:   END setTextString;
61:
62:  (* führt vollständigen Dialog aus *)
63:  PROCEDURE DoDialog (GEMDevice : DeviceHandle;
     tree : PtrObjTree) : CARDINAL;
64:
65:   VAR space      : Rectangle; (* Dialogbox-Größe  *)
66:       Status     : OStateSet; (* Object-Status    *)
67:       button     : CARDINAL;(* ausgewähltes Object*)
68:       MakeBits   : BOOLEAN; (* Flag für reserv.
                                  Speicher *)
69:       Picture    : POINTER TO ARRAY [0..32511] OF
                      CHAR;
70:       MemFSource,MemFDest : MemFormDef; (* FDB    *)
71:
72:   BEGIN
73:    UpdateWindow(TRUE);   (* keine Fenster-ausgaben
                               mehr *)
74:    SetCurrObjTree(tree,FALSE);
75:    space:=FormCenter(tree); (* Dialog zentrieren *)
76:    ALLOCATE(Picture,SIZE(Picture^));(* Platz für
                                              Bild     *)
77:    MakeBits:=Picture#NIL;            (* Platz OK?  *)
78:    IF MakeBits THEN
79:     GetLogMemForm(MemFSource);       (* FDB holen  *)
80:     MemFDest:=MemFSource;            (* Ziel-FDB   *)
81:     MemFDest.start:=(* Adresse auf volle 512 Byte*)
82:           ADDRESS(LONGCARD(Picture)+512L-(LONGCARD
                   (Picture) MOD 512L));
83:     GrafMouse(mouseOff,NIL); (* Maus aus    *)
84:     CopyOpaque(GEMDevice,  (* Hintergrund sichern *)
85:                ADR(MemFSource),ADR(MemFDest),
86:                space,space,onlyS);
87:     GrafMouse(mouseOn,NIL)      (* Maus wieder an  *)
88:    ELSE
89:     FormDial(reserveForm,Rect(0,0,0,0),space)
                             (* Bild ber. reserv.    *)
90:    END;
91:    DrawObject(tree,Root,MaxDepth,space); (* Dialog
                                               zeichnen *)
92:    FormDo(tree,Root,button);(*Dialog durchführen *)
93:    IF MakeBits THEN
94:     GrafMouse(mouseOff,NIL); (* Maus aus          *)
95:     CopyOpaque(GEMDevice, (* Hintergrund zurück   *)
96:                ADR(MemFDest),ADR(MemFSource),
97:                space,space,onlyS);
98:     GrafMouse(mouseOn,NIL);(* und wieder Maus an*)
99:     DEALLOCATE(Picture,SIZE(Picture^))(* Speicher
                                            freigeben  *)
100:   ELSE
101:    FormDial(freeForm,Rect(0,0,0,0),space);(* Bild.
                                          ber.freigeben  *)
102:   END;
103:   Status:=ObjectState(button); (* Status
                                  angeklicktes Object *)
104:   EXCL(Status,selectObj); (* SELECTED entfernen*)
105:   ChangeObjState(tree,button, (* neuen Status
                                    setzen          *)
106:                  TransRect(ObjectSpace(button),
107:                  ObjectOffset(tree,button)),
108:                  Status,FALSE);
109:   UpdateWindow(FALSE); (* Fensterausgabe
                            freigeben              *)
110:   RETURN button
111:  END DoDialog
112:
113: END DialogHandler.
114:
115: (* Listing 2b *)
```

Listing 2b

```
1: (*************************************************)
2: (* Definitionsmodul für MIDI-Routinen Version 0.0,
    19/09/88                                        *)
3: (*===============================================*)
4: (* Autor: Dietmar Rabich, Dülmen.Entw.mit Megamax
    Modula 2. *)
5: (*************************************************)
6: (* Erläuterungen befinden sich im
    Implementationsmodul.                           *)
7: (*************************************************)
8:
9:  DEFINITION MODULE MIDIHandler;
10:
11:  PROCEDURE SetTimeOut (t : CARDINAL);
12:
13:  PROCEDURE GetTimeOut : CARDINAL;
14:
15:  PROCEDURE MIDIReceived : BOOLEAN;
16:
17:  PROCEDURE WriteStringToMIDI (s : ARRAY OF CHAR);
18:
19:  PROCEDURE ReadStringFromMIDI(VAR s : ARRAY OF
     CHAR);
20:
21:  PROCEDURE WriteToMIDI (c : CHAR);
22:
23:  PROCEDURE ReadFromMIDI (VAR c: CHAR);
24:
25:  END MIDIHandler.
26:
27:  (* Listing 3a *)
```

Listing 3a

```
1:  (*********************************************)
2:  (* Implementationsmodul für MIDI-Routinen
                          Version 0.1, 27/09/88*)
3:  (*=========================================== *)
4:  (* Autor: Dietmar Rabich, Dülmen.
             Entw. mit Megamax Modula 2.*)
5:  (*********************************************)
6:
7:   IMPLEMENTATION MODULE MIDIHandler;
8:
9:   (* BIOS-Routinen *)
10:  FROM BIOS          IMPORT BConOut,BConIn,BConStat,
                               BCosStat,Device;
11:
12:  (* String-Routinen *)
13:  FROM FastStrings IMPORT Length;
14:
15:  (* AES-Routinen *)
16:  FROM AESEvents     IMPORT TimerEvent;
17:  FROM AESForms      IMPORT FormAlert;
18:
19:  (* TimeOut-Zeit in 1/100 s *)
20:  VAR TimeOut : CARDINAL;
21:
22:  (* setzt TimeOut in s *)
23:  PROCEDURE SetTimeOut (t : CARDINAL);
24:
25:   BEGIN
26:    TimeOut:=t*100
```

Listing geht weiter...

```
27:   END SetTimeOut;
28:
29:  (* gibt TimeOut in s zurück *)
30:  PROCEDURE GetTimeOut : CARDINAL;
31:
32:   BEGIN
33:    RETURN TimeOut DIV 100
34:   END GetTimeOut;
35:
36:(* Schreibt ein Zeichen über MIDI.(interne Routine)*)
37:(* Das Zeichen wird nur dannweitergegeben,wenn der*)
   (* Puffer nicht gänzlich gefüllt ist. Kann nach  *)
38:(* 5 s(=500*10 ms)noch immer kein Zeichen in den *)
39:(*  Puffer geschrieben werden, so gibt wird eine *)
40:(* Timeout-Meldung ausgegeben. Verläuft die      *)
   (* Übertragung korrekt, so ist die Rückgabe von  *)
41:(* WriteToMidi TRUE, sonst FALSE.                *)
42:(* 5 s ist die Standard-Vorgabe, die verändert
     werden kann.                                  *)
43:  PROCEDURE WriteToMidi (c : CHAR) : BOOLEAN;
44:
45:   VAR TimeCounter,VoidC : CARDINAL;
46:
47:   BEGIN
48:    TimeCounter:=0;
49:    WHILE ~BCosStat(HSS) AND (TimeCounter<500) DO
50:     TimerEvent(10L);
51:     INC(TimeCounter)
52:    END;
53:    IF TimeCounter<TimeOut THEN
54:     BConOut(HSS,c)
55:    ELSE
56:     FormAlert(1,'[0][MIDI - Timeout| ][ OK ]',VoidC)
57:    END;
58:    RETURN TimeCounter<TimeOut
59:   END WriteToMidi;
60:
61:  (* Analog zur WriteToMidi, allerdings werden die
                                  Zeichen gelesen. *)
62:  (* (interne Routine) *)
63:  PROCEDURE ReadFromMidi (VAR c : CHAR) : BOOLEAN;
64:
65:   VAR TimeCounter,VoidC : CARDINAL;
66:
67:   BEGIN
68:    TimeCounter:=0;
69:    WHILE ~BConStat(HSS) AND (TimeCounter<500) DO
70:     TimerEvent(10L);
71:     INC(TimeCounter)
72:    END;
73:    IF TimeCounter<TimeOut THEN
74:     c:=CHR(SHORT(BConIn(HSS) MOD $100L))
75:    ELSE
76:     FormAlert(1,'[0][MIDI - Timeout| ][ OK ]',VoidC
77:    END;
78:    RETURN TimeCounter<TimeOut
79:   END ReadFromMidi;
80:
81:  (* Status des MIDI-Eingangs                     *)
82:  (* TRUE, falls Zeichen verfügbar, FALSE sonst *)
83:  PROCEDURE MIDIReceived : BOOLEAN;
84:
85:   BEGIN
86:    RETURN BConStat(HSS)
87:   END MIDIReceived;
88:
89:  (* schreibt einen String über MIDI an den
     zweiten Rechner                              *)
90:  PROCEDURE WriteStringToMIDI (s : ARRAY OF CHAR);
91:
92:   VAR i,LengthOfMessage : CARDINAL;
93:       WriteFlag         : BOOLEAN;
94:       VoidCh            : CHAR;
95:
96:   BEGIN
97:    IF s[0]#0C THEN
98:     LengthOfMessage:=Length(s); (* Anzahl der
                                     Buchstaben  *)
99:     WriteFlag:=WriteToMidi(CHR(LengthOfMessage));
                         (* Länge übermitteln       *)
100:    i:=0;
101:    WHILE (i<LengthOfMessage) AND WriteFlag DO
                         (* String schreiben,       *)
102:     WriteFlag:=WriteToMidi(s[i]);
                         (*  falls bisher kein      *)
103:     INC(i)          (*  Timeout                *)
104:    END;
105:    IF WriteFlag THEN    (* Antwort abwarten     *)
106:     WriteFlag:=ReadFromMidi(VoidCh)
107:    END
108:   END
109:  END WriteStringToMIDI;
110:
111:(*liest einen String über MIDI vom zweiten Rechner*)
112:  PROCEDURE ReadStringFromMIDI (VAR s : ARRAY OF
                                          CHAR);
113:
114:  VAR Anzahl,i : CARDINAL;
115:      ReadFlag : BOOLEAN;
116:      Ch       : CHAR;
117:
118:  BEGIN
119:   ReadFlag:=ReadFromMidi(Ch);
120:   Anzahl  :=0;
121:   IF ReadFlag THEN
122:    Anzahl:=ORD(Ch);  (* Anzahl der Buchstaben  *)
123:    i:=0;
124:    WHILE (i<Anzahl) AND ReadFlag DO (* String
                                  lesen,falls    *)
125:     ReadFlag:=ReadFromMidi(s[i]); (*  kein
                                     Timeout    *)
126:     INC(i)
127:    END;
128:    IF ReadFlag THEN           (* Antwort senden *)
129:     ReadFlag:=WriteToMidi(0C)
130:    END
131:   END;
132:   s[Anzahl]:=0C           (* ASCII 0 anfügen  *)
133:  END ReadStringFromMIDI;
134:
135: (* schreibt einen Character über MIDI *)
136: PROCEDURE WriteToMIDI (c : CHAR);
137:
138:  VAR VoidB  : BOOLEAN;
139:      VoidCh : CHAR;
140:
141:  BEGIN
142:   IF WriteToMidi(c) THEN (* Character schreiben*)
143:    VoidB:=ReadFromMidi(VoidCh) (* Antwort
                                abwarten      *)
144:   END
145:  END WriteToMIDI;
146:
147: (* liest einen Character über MIDI *)
148: PROCEDURE ReadFromMIDI (VAR c: CHAR);
149:
150:  VAR VoidB : BOOLEAN;
151:
152:  BEGIN
153:   IF ReadFromMidi(c) THEN (* Character empfangen*)
154:    VoidB:=WriteToMidi(0C)     (* Antwort senden  *)
155:   END
156:  END ReadFromMIDI;
157:
158: BEGIN
159:  TimeOut:=500                 (* Voreinstellung   *)
160: END MIDIHandler.
161:
162: (* Listing 3b *)
```

Listing 3b

ENDE

Flexible Modulprogrammierung mit ADIMENS Talk

Teil 5

Ausgabemöglichkeiten der gefundenen Datensätze

Wie findet man die Daten, die man in der Datenbank sucht? Das war das Thema im Teil 4. Ich hoffe, daß Sie das vorgestellte Suchmodul für Ihren Anwendungsfall gebrauchen können.

Das naive Bild einer Datenbank reicht im allgemeinen bei einer zunehmenden Komplexität der Darstellung von Datenbeziehungen nicht aus. Deshalb möchte ich Ihnen in diesem Teil zuerst einige Grundlagen der verschiedenen Techniken der Datei-Adressierung aufzeigen. Anschließend wird anhand der Funktionsbeschreibung der folgenden vier Grundmoduln eine der vielfältigen Ausgabemöglichkeiten von gefundenen Datensätzen gezeigt werden. Die Vorstellung des Modulkonzeptes und die Beschreibung der Moduln mit Hilfe von Struktogrammen und von Hierachiediagrammen wird es Ihnen dann ermöglichen, Ihre Daten flexibel zu gebrauchen. Somit werden Sie zum Schluß die Möglichkeit haben, sämtliche verfügbaren Daten in überschaubarer und relativ einfacher Weise auszuwerten.

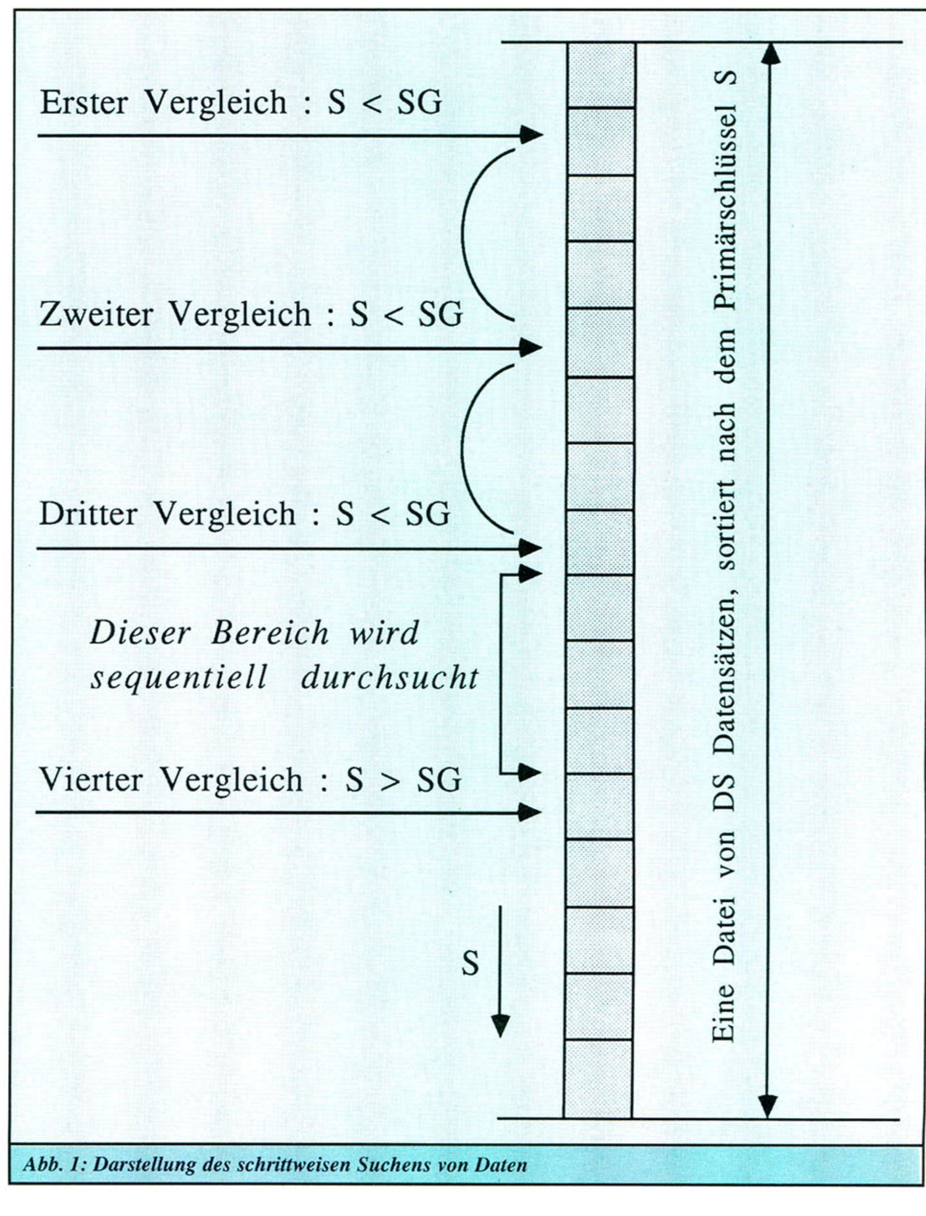


Abb. 1: Darstellung des schrittweisen Suchens von Daten

Techniken zur Lokalisierung eines Datensatzes

a) Das sukzessive Durchsuchen der Datei

Dies ist wohl die einfachste, aber auch grobschlächtigste Methode, einen Datensatz innerhalb einer Datei zu lokalisieren. Der Datensatzzeiger wird auf den Anfang

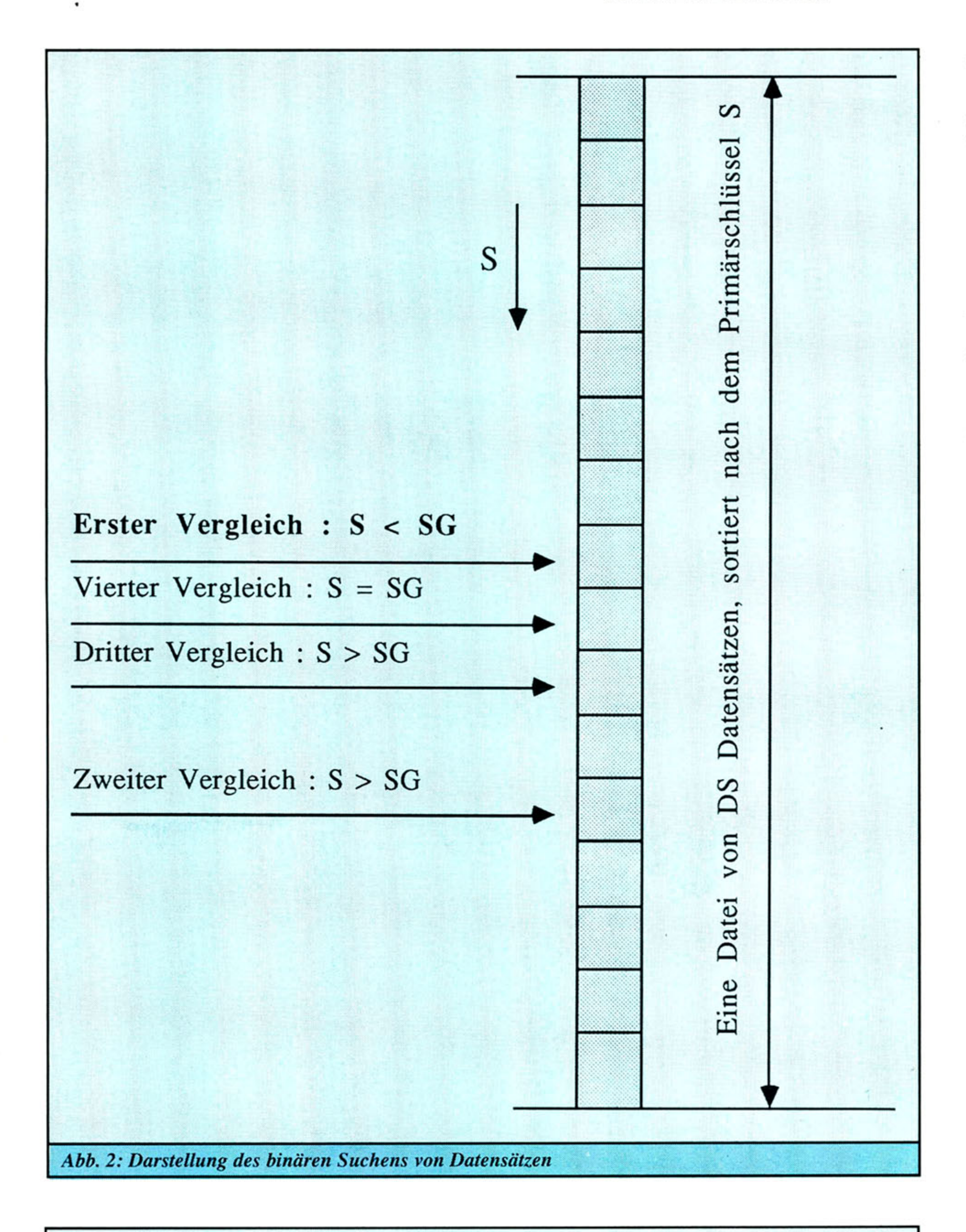


Abb. 2: Darstellung des binären Suchens von Datensätzen

```
===============================================================================
 Datum                  DATEN - SUCHE              *** V 1.01 ***      Uhrzeit
 16.07.88    flexible Anwendungsprogrammierung    mit ADIMENS Talk     18:11:28
-------------------------------------------------------------------------------

                       SUCHEN  neuer Lieferanten ADRESSEN
===============================================================================


  Bitte geben Sie einen Zahlenbereich an.           Letzte Lieferantennr.:10
                                                    Akt. Lieferantennr.  :10
  Unterer Zahlenwert :         3                    Anzahl Datensätze    :10
  Oberer Zahlenwert  :        10

===============================================================================
  SUCHE im Bereich von : 3 bis 10                     BITTE WARTEN ...........
===============================================================================
                       Es wurden 8 Datensätze gefunden
===============================================================================
BILDSCHIRM Adressen <F1>  DATENSÄTZE alle <F5>    DRUCKER <F7>       <F10>
 AUSDRUCK  Telefon  <F2>    DRUCKEN  best.<F6>  INITIALISIEREN       ENDE
```

Abb. 3: Differenzierte Suche nach DS im Bereich der L.-Nr. von 3 bis 10

der Datei gesetzt. Der Zeiger rückt dann jedesmal nur um einen Datensatz weiter und vergleicht dabei die Merkmalsausprägung des zu suchenden Datensatzes.

Diese Methode ist für die meisten Zwekke viel zu langsam und kann wahrscheinlich nur bei kleinen Satzmengen benutzt werden. Im letzten Teil wurde deshalb bei der Suche nach einem Lieferantennamen in einem bestimmten Bereich von Lieferantennummern eine Sicherheitsabfrage bei der sukzessiven Suche von mehr als 100 Datensätzen implementiert.

b) Das schrittweise Suchen von Datensätzen

Wenn die Datensätze sequentiell geordnet, also nach einem Schlüssel organisiert sind, braucht nicht jeder Satz gelesen zu werden, um die Datei zu duchsuchen. Der Rechner kann z.B. jeden hundertsten Satz in aufsteigender Folge des Schlüssels untersuchen (siehe Abb. 1). Wenn ein Satz gefunden wird, dessen Schlüsselwert größer als der gesuchte Schlüsselwert ist, werden die 99 zuletzt übersprungenen Sätze sequentiell durchsucht.

Diese Technik wird *"das schrittweise Suchen"* genannt, wobei die Schrittweite zwar konstant aber wählbar sein muß. Dies könnte eine Optimierungsmethode bei der Suche nach zwei Bedingungen sein. Angenommen die Anzahl der Sätze in Ihrer Datei, die durchsucht werden soll, wird mit *DS* bezeichnet, dann beträgt die optimale Schrittweite $OS = \sqrt{DS}$. In jedem Schritt über *OS* Datensätze hinweg wird nur einmal ein Schlüsselvergleich durchgeführt, bis die zutreffende Gruppe von Sätzen erreicht ist.

Das schrittweise Suchen ist für das Durchsuchen sehr großer Dateien ungeeignet. Es ist aber eine wichtige Technik zum Suchen in einem relativ kleinen Ausschnitt aus einer Datei.

c) Das binäre Suchen

Das binäre Suchen kann angewendet werden, wenn die Sätze nach ihrem Schlüssel sortiert sind. Das binäre Suchen beginnt am Mittelpunkt des zu durchsuchenden Bereiches und vergleicht den Schlüssel des dortigen Satzes mit dem gesuchten, womit einer der beiden Halbbereiche als zutreffend erkannt wird. Dann wird dieser Bereich halbiert und der gleiche Prozeß wiederholt (s. Abb. 2). Die Halbierung des Bereiches wird so fortgesetzt, bis der gewünschte Satz gefunden oder als nicht in der Datei enthalten erkannt ist. Wie das schrittweise Suchen ist auch das binäre Suchen eher für das Suchen in kleineren Ausschnitten von Dateien geeignet.

d) Die Suche nach einem Index

Indizieren gibt ihrer Datei den Anschein, als wäre sie sortiert, ohne die Reihenfolge

der Datensätze wirklich zu verändern. ADIMENS erstellt eine Indexdatei mit dem Befehl *INDEX > Merkmalsname <*. In der Indexdatei, die als Binärbaum organisiert ist, befinden sich nur die Inhalte des Feldes, nach dem indiziert worden ist, der sogenannte Schlüssel. Die *Bayer-Bäume* sind dabei eine spezielle Art dieser Binärbäume. In jedem Knoten sind mehrere Verweise (ca. 20 - 50) auf weitere Knoten oder auf Blätter. Verweise auf echte Blätter und die dort gespeicherten Datensätze findet man nur in den untersten Knoten. Alle Knoten haben die gleiche Größe (z.B. 1 KByte) und lassen sich daher gut an die Sektorgröße einer Diskette oder Festplatte anpassen. Datenbanken mit Bayer-Baum-Struktur zeichnen sich durch besonders kurze Zugriffzeiten aus. ADIMENS verdankt seine hohe Arbeitsgeschwindigkeit derartigen Algorithmen, die optimal auf Bayer-Bäume zugeschnitten sind. So erreicht man kurze Zugriffszeiten auf alle Datensätze. Die Suchzeiten hängen im wesentlichen von der Höhe des Datenbaumes ab. Auch bei mehreren tausend Datensätzen sind die Bayer-Bäume von ADIMENS zum Suchen eines beliebigen Datensatzes nur unwesentlich länger als zum Zugriff auf den ersten. Die Reihenfolge der Schlüsselbegriffe in der Indexdatei bestimmt die Reihenfolge, in der auf die Daten in Ihrer Datei zugegriffen wird. Diese Methode ermöglicht einen wesentlich schnelleren Zugriff auf die Daten, als es bei einer nichtindizierten Datei möglich wäre.

Kombination von Techniken

Zusammenfassend möchte ich hier noch einmal betonen, daß die Datensuche vom Datenbanksystem ADIMENS sehr gut unterstützt wird. Leider tauchen aber in der Praxis Probleme auf, die man mit Hilfe der Kenntnis dieser Suchtechniken besser, schneller und effizienter lösen kann.

Bei manchen Problemen können Sie Kombinationen der oben genannten Techniken benutzen, um die Sätze zu adressieren. Ein Index z.B. kann einen Bereich der Datei lokalisieren, und dieser wird dann sequentiell oder nach der Binär-Suchmethode durchsucht. Die Suche nach einem Lieferantennamen in einem bestimmten Bereich der Lieferantennummern können Sie sehr elegant mit Hilfe dieser Methode lösen.

```
================================================================================
Datum                DATEN - SUCHE                   *** V 1.01 ***      Uhrzeit
16.07.88     flexible Anwendungsprogrammierung     mit ADIMENS Talk     18:19:06
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

            Anzeige von Datensätzen - ADRESSEN LISTE

================================================================================

                     Lieferanten Nr. : 3

                     Firma : Huber GmbH
                     Name  : Herbert Huber
                     Sieglindenstr. 5
                     8000 München

================================================================================

            E = BEENDEN
                   WEITERMACHEN, BELIEBIGE TASTE DRÜCKEN
```

Abb. 4: Anzeige einer Adresse auf dem Bildschirm

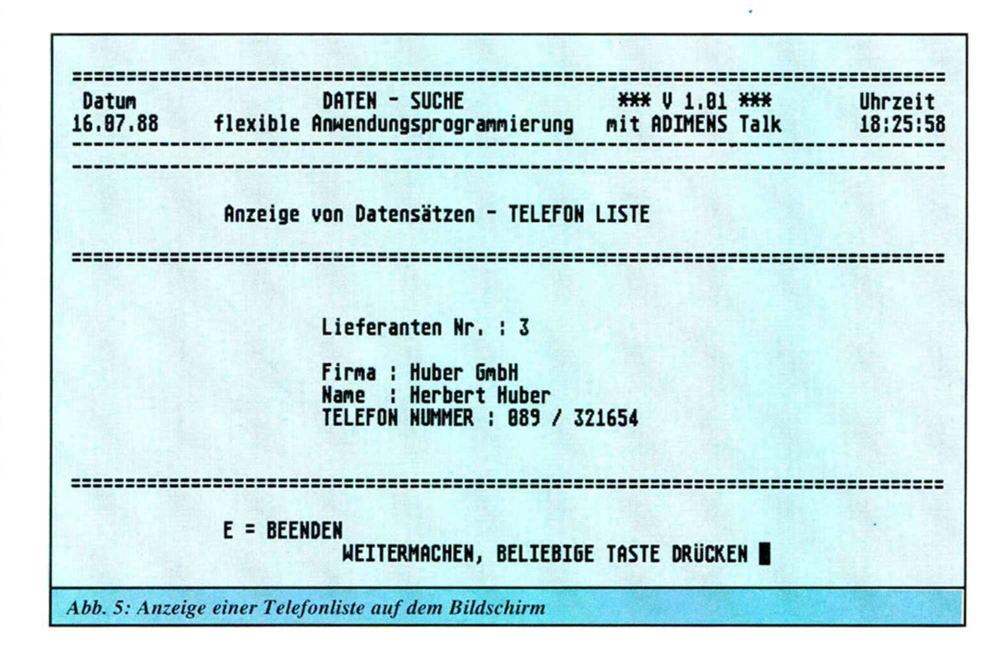


Abb. 5: Anzeige einer Telefonliste auf dem Bildschirm

```
================================================================================
Datum                DATEN - SUCHE                   *** V 1.01 ***      Uhrzeit
16.07.88     flexible Anwendungsprogrammierung     mit ADIMENS Talk     18:34:54
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

                     DRUCKEN von Datensätzen

================================================================================
 Die gefundenen Datensätze werden alle in eine              DRUCKEN
         temporäre Datei kopiert.
                                                   Bitte Papier einlegen und
 Anzahl der zu kopierenden Datensätze : 8          Drucker eingeschalten ???
 schon kopierte Datensätze            : 07
================================================================================

================================================================================
 Kopierstart                   BITTE WARTEN !!!
================================================================================
```

Abb. 6: Kopieren der DS in eine temporäre Datei

Funktionsbeschreibung der Module

Das Suchen nach Datensätzen mit speziellen Merkmalsausprägungen sollte genau auf den Anwendungsfall hin geplant werden. Außerdem sollte die Möglichkeit der differenzierten Selektion von Datensätzen gegeben sein.

Im letzten Teil wurden deshalb zwei Module vorgestellt, die die benutzerfreundliche Suche von Datensätzen in einem Anwendungsprogramm unterstützen. Man hatte dort die Möglichkeit, nach einem numerischen und einem alphanumerischen Merkmal zu suchen.

In Abb.3 sehen Sie den Bildschirmausdruck bei einer differenzierten Suche in einem Bereich von Lieferantennummern. Da die Anzahl der Datensätze mit der letzten Lieferantennummer übereinstimmt, sind keine Datensätze im gewählten Bereich gelöscht worden. Deshalb wurden auch 8 davon gefunden.

Nun haben Sie die Möglichkeit, verschiedene Optionen auszuwählen. Für diese Auswahloptionen sollen, in diesem Teil der Serie, flexible, universell einsetzbare Module aufgezeigt werden.

Zuerst möchte ich Ihnen aber die einzelnen Module vorstellen. Der Anwender hat die Möglichkeit, verschiedene Fälle mit den Funktionstasten anzuwählen, nachdem die Datensätze gefunden wurden. Mit der Tastenbetätigung F10 wird eine Adresse auf dem Bildschirm angezeigt (siehe Abb. 4). Dabei wurde nach der Lieferantennummer aufsteigend sortiert. Die Sortierrichtung kann vom ADI-Talk-Menü oder im Quelltext mit dem *SET*-Befehl umgestellt werden.

Wenn mit der Funktionstaste F1 der gefundene Datensatz angezeigt wurde, kann mit "E" wieder ins Menü >*Suchen nach Datensätzen*< gesprungen werden. Ansonsten können Sie mit einer beliebigen Taste "weiterblättern". Der Datensatzzeiger wird dabei um Eins erhöht und der nächste Datensatz, der den Bedingungen genügt, angezeigt. Diese Möglichkeit haben Sie natürlich nur solange, bis das Ende Ihrer Datei erreicht ist, oder bis alle gefundenen Datensätze auf dem Bildschirm angezeigt wurden.

Mit der Funktionstaste F2 können Sie sich von diesem Datensatz die Telefonliste anzeigen lassen (Abb. 5). Diese Unterscheidungen werden von Fall zu Fall unterschiedlich sein. Trotzdem sollten

```
================================================================================
Datum              DATEN - SUCHE                  *** V 1.01 ***         Uhrzeit
16.07.88   flexible Anwendungsprogrammierung   mit ADIMENS Talk         18:34:54
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------

                      DRUCKEN von Datensätzen

================================================================================
             Bitte entscheiden Sie sich für einen Sortierschlüssel
             es wird dabei jeweils aufsteigend sortiert.

             sortieren nach dem LIEFERANTEN NAMEN : --> 'N' drücken
             sortieren nach der LIEFERANTEN NUMMER: --> 'L' drücken
                                  EINGABE: N
================================================================================
```

Abb. 7: Sortiermerkmale auswählen

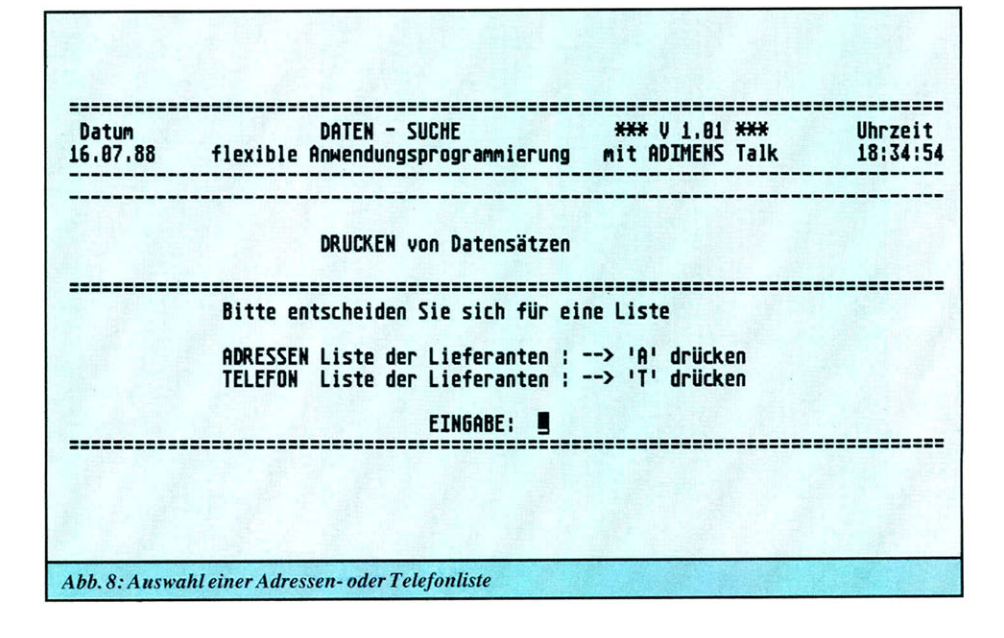


Abb. 8: Auswahl einer Adressen- oder Telefonliste

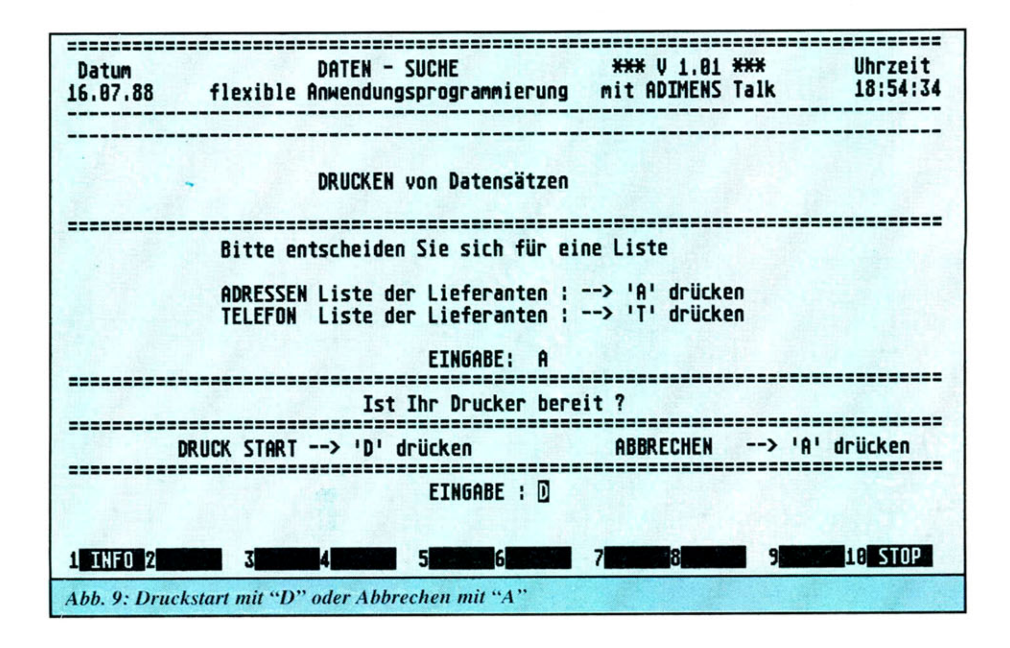


Abb. 9: Druckstart mit "D" oder Abbrechen mit "A"

Sie bedenken, daß man auch hier eine Möglichkeit hat, bestimmte Informationen vom Zugangsberechtigungscode des jeweiligen Benutzers abhängig zu machen.

Sie haben mit dem Anwendungsprogramm immer ein Ziel. Zur Erfüllung der erforderlichen Aufgaben (vgl. Teil 2 - konzeptionelle Schritte bei der Einführung einer Datenbank) müssen vom Datenbanksystem ADIMENS Informationen von außen aufgenommen, verarbeitet und entsprechende Informationen nach außen wieder abgegeben werden. Durch die Integration aller Ihrer Daten können Sie sie nach anderen Gesichtspunkten als bisher vorgesehen auswerten.

Deshalb werden die nachfolgenden Module es Ihnen erlauben, spezielle Sichten von Benutzergruppen auf die Daten abzuändern. Damit können Sie neue Sichten für Personengruppen einführen.

Für die meisten Anwendungen genügt es allerdings nicht, nur die Daten auf dem Bildschirm auszugeben, sondern man möchte auch Datensätze auf dem Drucker ausgeben lassen, beispielsweise, um eine sortierte Telefonliste oder eine nach bestimmten Merkmalen ausgewählte Adressenliste zu erstellen.

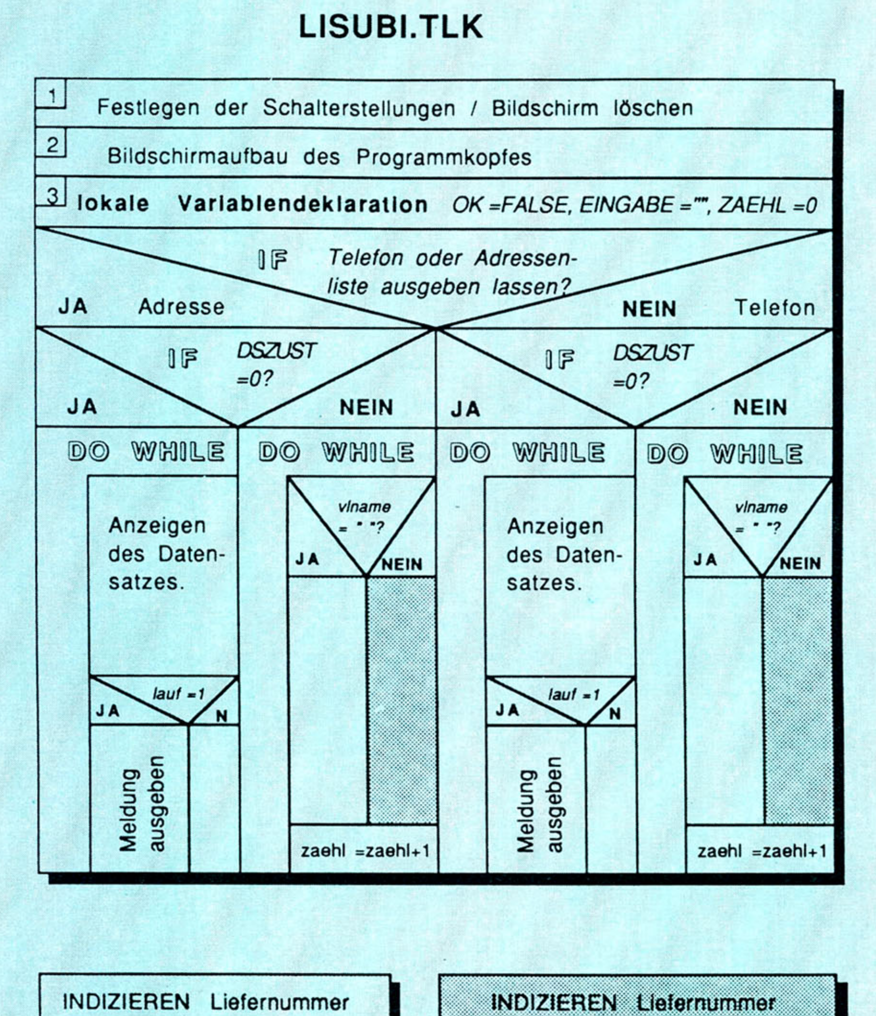
Für diese Zwecke benötigen Sie ein Modul, welches die Datensätze in eine temporäre Datei kopiert. Diese Datei hat die gleichen Merkmalsnamen wie die aktuelle Datei. Wollen Sie eine Adressenliste ausdrucken lassen, werden zunächst alle gefundenen Datensätze in eine temporäre Datei kopiert. Diese kann dann nach den verschiedensten Merkmalen indiziert werden. Außerdem können Sie die Datei mehrmals hintereinander ausdrucken lassen.

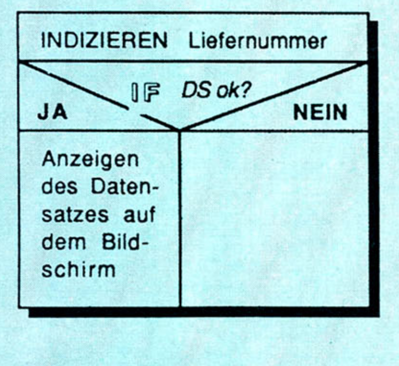
In der Zeit, in der die Datensätze kopiert werden, können Sie Ihren Drucker einschalten und das Papier einlegen. Abb. 6 zeigt diese Phase. Nachdem das Kopierende erreicht ist, erscheint die Meldung, daß man eine beliebige Taste drücken soll, um weiterzumachen.


ADRESSEN LISTE vom : 16.07.88 SEITE: 1

NR	FIRMA	STRASSE	PLZ	WOHNORT
9	Helena Busch, Busch & Partner	Hainallee 86,	4600	Dortmund
3	Herbert Huber, Huber GmbH	Sieglindenstr. 5,	8000	München
6	Klaus Knoll, Knoll & Söhne	Amselweg 23,	1000	Berlin
8	Ludwig Krause, Sport Krause	Silber Strasse 12,	4300	Essen
7	Markus Weber, Weber - KG	Olympia Strasse 44,	8000	München
5	Peter Hoffmann, Hoffmann OHG	Alte Gasse 15,	8630	Coburg
10	Stephan Schulte, Dr.Schulte	Ostwall 4,	4400	Münster

Abb. 10: Beispielausdruck einer Adressenliste





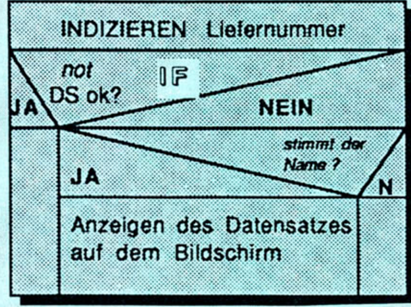


Abb. 12: Struktogramm von LISUBI.TLK

In den nachfolgenden beiden Menüs (s. Abb. 7 und Abb. 8) können Sie verschiedene Sortiermerkmale auswählen und zwischen einer Adreßliste oder einer Telefonliste wählen. In unserem Beispiel sollte nach dem Lieferantennamen sortiert und eine Adressenliste erstellt werden.

Aus Abb. 9 wird ersichtlich, daß man zu diesem Zeitpunkt noch die Möglichkeit hat, den Druckmodus zu verlassen. Das Modul zum Drucken von Datensätzen ist eingestellt auf eine Zeilenanzahl von 64 pro Seite. Nachdem 64 Zeilen (= Datensätze) ausgedruckt wurden, soll jedesmal eine Abfrage ausgegeben werden, ob man den Druck abbrechen möchte. Außerdem soll auf jeder neuen Seite die aktuelle Seitenanzahl stehen und die Kopfzeile jedesmal am Anfang ausgedruckt werden. Bei einer großen Adressenliste ist es immer ärgerlich, wenn die Kopfzeile der Liste nur einmal am Anfang steht.

Zudem kann der Fall auftreten, daß während des Druckens Ihr Papier zu Ende geht. Deshalb ist es sinnvoll, das Ausdrucken Ihrer Liste zum Beginn einer neuen Seite zu unterbrechen.

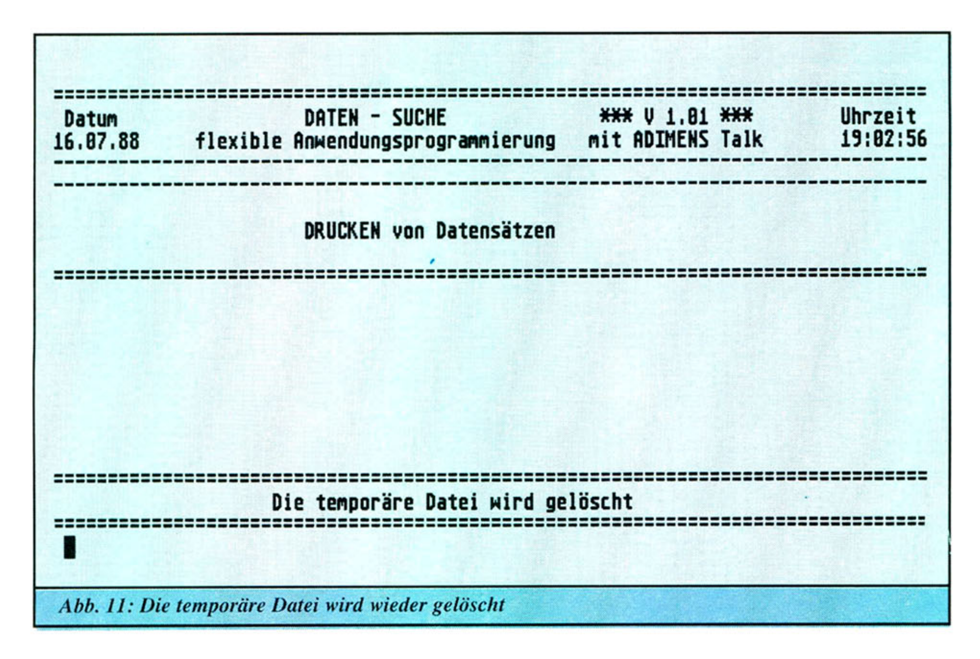


Abb. 11: Die temporäre Datei wird wieder gelöscht

Einen Beispielausdruck einer kleinen Liste zeigt Abb. 10.

Nachdem die gewünschte Liste ordnungsgemäß ausgedruckt wurde, kann das Programm mit "A" (Abbruch) abgebrochen werden. Mit der Eingabe "D" (Druckstart) wird die Liste dupliziert. Beim Programmabbruch wird die vorher erstellte temporäre Datei wieder gelöscht (s. Abb. 11). Nun befindet man sich erneut im Menü Suchen von Datensätzen.

Vorstellung der Modulkonzeption

Ich werde Ihnen zwei Ausgabemöglichkeiten - die Bildschirmausgabe und die Ausgabe auf einem Drucker - mit ADIMENS Talk aufzeigen. Das Modulstruktur-Diagramm zeigt dabei die Zusammenhänge der Module vom Teil 4 und von diesem Teil. Kurz zur Wiederholung: Das Modul LSUCHE.TLK stellt die Suchmaske und die Verzweigungsmöglichkeiten zur Verfügung. Mit dem Modul LNRBER.TLK wird die Eingabe einer Nummer iterativ überarbeitet. In diesem Teil sind die vier Module LISUBI.TLK, LISUDR.TLK, LCOPY.TLK und LDRUCKAD.TLK. neu hinzugekommen. Die Module LCOPY und LDRUCKAD werden wiederholt vom Modul LISUDR aufgerufen und abgearbeitet. LCOPY kopiert die spezifischen Lieferantendatensätze in die temporäre Datei (s. auch Teil 2: Abb. 5 - Allgemeiner Aufbau eines Modulstruktur-Diagrammes). LDRUCKAD stellt sozusagen das Ausgabeformat der auszudruckenden Datensätze zur Verfügung und regelt die Druckausgabe mit der Sicherheitsabfrage vor jeder neuen Seite.

Damit die temporäre Datei gelöscht wird, mußte ich dem Modul LSUCHE noch einen Anweisungsblock hinzufügen. Das Listing 1 zeigt diesen zusätzlichen Anweisungsblock. Sie können Block 11 direkt hinter den Block 10 einfügen, der im Teil 4 (Modul LSUCHE) vorgestellt wurde.

Zusammenfassung Mit den Teilen 4 und 5 wurde eine der Möglichkeiten aufgezeigt, Moduln für das Suchen von Daten auf systematische Art und Weise zu gestalten. Neben diesen Hauptaufgaben - Maskenaufbau, Datensicherung, Erfassen und Suchen von Datensätzen, die immer bei der Erstellung einer Datenbank-Anwendungsprogrammierung auftreten, fehlt noch ein weiteres wichtiges Modul, das das Ändern und das Löschen von Datensätzen aus einer bestimmten Datei gewährleistet.

Von diesem voraussichtlich letzten universell einsetzbaren Modul wird Teil 6 in der nächsten Ausgabe berichten.

Hans-Ulrich Mayer

```
 1: /**** 11 ****/
 2: /* Die temporäre Kopierdatei wird wieder gelö.   */
 3: /* Es gibt bei ADIMENS keinen Befehl, bei dem die */
 4: /* ganze Datei gelöscht wird. Deshalb muß man     */
 5: /* sukzessive jeden einzelnen Datensatz löschen.  */
 6:  IF TEMP
 7:     § 18,00 CLEAR
 8:     §18,00 SAY BILD1 + BILD1
 9:     § 19,20 SAY "Die temporäre Datei wird gelöscht"
10:     § 20,00 SAY BILD1 + BILD1
11:     STORE 0 TO ZAEHL
12:     USE ADRESSENCOPY
13:     INDEX LNUMMER
14:
15: /************************************************/
16: /*SOLANGE die Zählvariable ungleich der Zahl der */
17: /*gefundenen Datensätze ist. Die Datens. werden  */
18: /*einzeln gelöscht.                              */
19: DO WHILE NOT (ZAEHL = GD)
20: /************************************************/
21:
22:        /* Der erste Datensatz muß immer fest
              bleiben, */
23:        /*  da die Datei indiziert wurde.              */
24:         SKIP 1
25:         DELETE
26:         STORE ZAEHL + 1 TO ZAEHL
27:     /* BEENDEN der DO-WHILE-Schleife                  */
28:     ENDDO
29:
30:     USE ADRESSEN
31:  ELSE
32:     /* NICHTS TUN                                     */
33:  ENDIF
34:  /**** 11 ****/
```

Listing 1: Zusätzlicher Anweisungsblock zum Löschen einer temporären Datei

```
 1: /*   PROGRAMM LISUBI.TLK                          */
 2: /**********************************************/
 3: /* Dies ist ein Beispiel für ein Programmodul    */
 4: /* zum Ausdrucken von Datensätzen innerhalb der  */
 5: /* Datenbank Das Programm greift auf die Datenb. */
 6: /* BEISPIEL zu.                                  */
 7: /* copyright by Hans-Ulrich Mayer                */
 8: /**********************************************/
 9:
10: /**** 1 ****/
11: /* FESTLEGUNG DER SCHALTERSTELLUNGEN             */
12: /* Löschen des Bildschirmes                      */
13:  CLEAR
14: /* Unterdrücken der automatischen Maskenanzeige  */
15:  SET AUTOSC OFF
16: /* Die Sonderregelungen für die Funktion FIND sind
       abgeschaltet                                  */
17:  SET DBASE OFF
18: /* Die Datenbank und die Datei wurde schon geöf. */
19: /**** 1 ****/
20:
21: /**********************************************/
22:
23: /**** 2 ****/
24: /* MENÜAUSGABE - Aufbau der Kopfzeile            */
25: CLEAR
26: /* aktuelle Datumsanzeige und der Uhrzeit        */
27:  @ 02     ,00  SAY " Datum "
28:  @ 03     ,00  SAY DATE()
29:  @ 02     ,72  SAY "Uhrzeit"
30:  @ 03     ,72  SAY TIME()
31:  @ 01     ,00  SAY BILD1 + BILD1
32:  @ ROW()  ,TAB SAY "          DATEN - SUCHE
        *** V 1.01 *** "
33:  @ ROW()+1,TAB SAY "flexible Anwendungs-
                     programmierung mit ADIMENS Talk"
34:  @ ROW()+1,00  SAY BILD2 + BILD2
35:  @ ROW()  ,00  SAY BILD2 + BILD2
36:  @ ROW()+1,00  SAY " "
37:  IF ( MSG = "F01" )
38:      @ ROW(),TAB SAY " Anzeige von Datensätzen -
         ADRESSENLISTE "
39:  ELSE
40:      @ ROW(),TAB SAY " Anzeige von Datensätzen -
         TELEFONLISTE  "
41:  ENDIF
42:  @ ROW()+1, TAB SAY " "
43:  @ ROW()+1,00 SAY BILD1 + BILD1
44:  /**** 2 ****/
45:
46:  /*********************************************/
47:
48:  /**** 3 ****/
49:  /* LÖSCHEN des mittleren Bildschirmbereiches    */
50:  /* Durchlaufe die Schleife SOLANGE die Eingabe
        richtig ist.                                 */
51:  @ 10,00 CLEAR
52:  /* Lokale Variablendeklaration                  */
53:  STORE FALSE TO OK
54:  STORE SPACE(1) TO EINGABE
55:  STORE 0 TO ZAEHL
56:  /**** 3 ****/
57:
58:  /*********************************************/
59:
60:  /* UNTERSCHEIDUNG zwischen einer Adressenliste und
        einer Telefonliste                           */
61:  IF ( MSG ="F01" )
62:
63:  /* FALL, für das Anzeigen einer Adressenliste
          auf dem Bildschirm                         */
64:  /* Wenn keine L.-Nummer eingegeben wurde,
          Suche nachdem L.- Namen                    */
65:     IF ( DSZUST = "0" )
66:
67:        /**** 4 ****/
68:  /* Es wird der Datensatz geladen, welcher die Bed.
      erfüllt.                                       */
69:        STORE ( JUMP VLNAME TO LNAME ) TO OK
70:
71:  /*********************************************/
72:  /* SOLANGE DURCHBLÄTTERN bis die Suche beendet */
73:  /* wird oder bis alle Datensätze mit der        */
74:  /* angegebenen Bedingung gezeigt wurden.        */
75:       DO WHILE NOT ( (ZAEHL = GD) OR (EINGABE = "E"))
76:  /*********************************************/
```

```
 77:
 78: /* Es werden hier nur zwei verschiedene Fälle
        unterschieden                                */
 79:            STORE 0 TO LAUF
 80:            @ 12,00 CLEAR
 81: /* ANZEIGEN des Datensatzes                     */
 82:            @ 12,23 SAY "Lieferanten Nr. : " + TRIM
                (STR(LNUMMER,6,0))
 83:            @ 14,23 SAY "Firma : "          + TRIM
                (LFIRMA)
 84:            @ 15,23 SAY "Name  : "          + TRIM
                (LNAME)
 85:            @ 16,23 SAY LSTRASSE
 86:            @ 17,23 SAY STR(LPLZ,4,0)       + " " +
                LORT
 87:            @ 19,00 SAY BILD1 + BILD1
 88:            STORE 1 TO LAUF
 89:            IF ( LAUF = 1 )
 90:               @ 21,13 SAY " E = BEENDEN "
 91:               @ 22,25 WAIT "WEITERMACHEN BELIEBIGE
                         TASTE DRÜCKEN" TO EINGABE
 92:            ENDIF
 93:            STORE ZAEHL + 1 TO ZAEHL
 94:            SKIP 1
 95: /* BEENDEN der Do-While SCHLEIFE                */
 96:        ENDDO
 97:
 98: /* ZEIGER auf den ersten Datensatz zurücksetzen und
        Entfernen des                                */
 99: /* Suchschlüssels.                              */
100:        BACK
101:        /**** 4 ****/
102:
103:     ELSE
104:
105: /**********************************************/
106: /* Fall, bei dem unter Umständen mehrere        */
107: /* Datensätze gezeigt werden. Solange bis alle */
108: /* gefunden Datensätze angezeigt wurden oder    */
109: /* durch eine Eingabe bestätigt wird, daß keine*/
110: /* Datensätze mehr angezeigtwerden sollen.      */
111:    DO WHILE NOT ( (ZAEHL = GD) OR (EINGABE = "E") )
112: /**********************************************/
113:
114:            STORE STR(DSZU,6,0) TO DSZUST
115:            STORE 0 TO LAUF
116:  /* FALL, bei dem kein Namen angegeben wurde    */
117:  /* und bei dem nur nach der Lieferanten        */
118:  /* Nummer gesucht werden soll.                 */
119:            IF ( VLNAME = "" )
120:
121:               /**** 5 ****/
122:               INDEX LNUMMER
123:               STORE ( FIND INT(DSZUST) ) TO OK
124:               IF (OK)
125:  /* ANZEIGEN DES DATENSATZES                    */
126:                  @ 12,00 CLEAR
127:                  @ 12,23 SAY "Lieferanten Nr. : " +
                      TRIM(STR(LNUMMER,6,0))
128:                  @ 14,23 SAY "Firma : "          +
                      TRIM(LFIRMA)
129:                  @ 15,23 SAY "Name  : "          +
                      TRIM(LNAME)
130:                  @ 16,23 SAY LSTRASSE
131:                  @ 17,23 SAY STR(LPLZ,4,0)       +
                      " " +LORT
132:                  @ 19,00 SAY BILD1 + BILD1
133:                  STORE 1 TO LAUF
134:                  STORE ZAEHL+1 TO ZAEHL
135:               ENDIF
136:               /**** 5 ****/
137:
138:            ELSE
139:
140:               /**** 6 ****/
141: /* Fall, bei dem nach der Lieferanten           */
142: /* Nummer und nach dem Namen des                */
143: /* Lieferanten gesucht werden soll              */
144:               INDEX LNUMMER
145:               STORE ( FIND INT(DSZUST) ) TO OK
146:               IF (OK)
147:                  IF ( LNAME = VLNAME )
148: /*ANZEIGEN des aktuellen Datensatz              */
149:                     @ 12,00 CLEAR
150:                     @ 12,23 SAY "Lieferanten Nr.:"+
                         TRIM(STR(LNUMMER,6,0))
```

Listing geht weiter...

```
151:                  @ 14,23 SAY "Firma : "         +
                            TRIM
                          (LFIRMA)
152:                  @ 15,23 SAY "Name   : "        +
                            TRIM(LNAME)
153:                  @ 16,23 SAY LSTRASSE
154:                  @ 17,23 SAY STR(LPLZ,4,0)      +
                            LORT
155:                  @ 19,00 SAY BILD1 + BILD1
156:                  STORE 1 TO LAUF
157:                  STORE ZAEHL + 1 TO ZAEHL
158:               ENDIF
159:            ENDIF
160:            /**** 6 ****/
161:
162:         ENDIF
163:         IF ( LAUF = 1 )
164:            @ 21,13 SAY  " E = BEENDEN "
165:            @ 22,25 WAIT "WEITERMACHEN, BELIEBIGE
                      TASTE DRÜCKEN" TO EINGABE
166:         ENDIF
167: /*DATENSATZZEIGER um einen Datensatz erhöhen    */
168:         STORE DSZU+1 TO DSZU
169:
170: /* BEENDEN der Schleife für die Anzeige v. Datens.*/
171:      ENDDO
172: /***********************************************/
173:
174:    ENDIF
175: /* BEENDEN des Falles für die Bildschirm Anzeige v.
   Datensätzen                                      */
176:
177:  ELSE
178:
179: /* FALL, für die Anzeige einer Telefonliste auf */
180: /* dem Bildschirm. Wenn keine Lieferanten Nr.   */
181: /* eingegeben wurde ->Suche nach dem Namen      */
182:    IF ( DSZUST = "0" )
183: /* Es wird der Datensatz geladen, welcher die Bed.
   erfüllt.                                         */
184:      STORE ( JUMP VLNAME TO LNAME ) TO OK
185:
186: /***********************************************/
187: /* SOLANGE die Bildschirmausgabe nicht beendet  */
188: /* werden soll, oder nicht alle gefundenen      */
189: /* Datensätze gefunden worden sind.             */
190:      DO WHILE NOT((ZAEHL = GD) OR (EINGABE =
                                        "E") )
191: /***********************************************/
192:
193:         /**** 7 ****/
194: /* Es werden hier zwei Fälle unterschieden      */
195:         STORE 0 TO LAUF
196:         @ 12,00 CLEAR
197: /* ANZEIGEN dieses Datensatzes                  */
198:         @ 12,23 SAY "Lieferanten Nr. : " + TRIM
                   (STR(LNUMMER,6,0))
199:         @ 14,23 SAY "Firma : "          + TRIM
                   (LFIRMA)
200:         @ 15,23 SAY "Name   : "         + TRIM
                   (LNAME)
201:         @ 16,23 SAY "TELEFON Nummer : " +
                   LTELEFON
202:         @ 19,00 SAY BILD1 + BILD1
203: /* Zuordnung einer Lieferanten Nummer           */
204:         STORE 1 TO LAUF
205:         IF ( LAUF = 1 )
206:            @ 21,13 SAY  "E = BEENDEN"
207:            @ 22,25 WAIT "WEITERMACHEN, BELIEBIGE
                      TASTE DRÜCKEN" TO EINGABE
208:         ENDIF
209:         STORE ZAEHL+1 TO ZAEHL
210: /* DATENSATZ ZEIGER um eins erhöhen             */
211:         SKIP 1
212:         /**** 7 ****/
213:
214: /* BEENDEN der Schleife                         */
215:      ENDDO
216: /***********************************************/
217:
218: /* ZEIGER auf den ersten Datensatz zurücksetzen */
219: /* und Entfernen des  Suchschlüssels.           */
220:      BACK
221:
222:    ELSE
223:
224: /***********************************************/
225: /* FALL bei dem unter Umständen mehrere         */
226: /* Datensätze angezeigt werden SOLANGE alle     */
227: /* gefundenen Datensätze angezeigt werden oder  */
228: /* durch eine Eingabe bestätigt wird, daß keine */
229: /* Datensätze mehr angezeigt werden sollen.     */
230:      DO WHILE NOT ( (ZAEHL = GD) OR (EINGABE = "E")
231: /***********************************************/
232:
233:         STORE STR(DSZU,6,0) TO DSZUST
234:         STORE 0 TO LAUF
235: /* FALL, bei dem kein Namen angegeben wurde     */
236: /* und nur nach der Lieferanten Nummer          */
237: /* gesucht werden soll.                         */
238:         IF (VLNAME = " ")
239:
240:            /**** 8 ****/
241:            INDEX LNUMMER
242:            STORE FIND INT(DSZUST) TO OK
243:            IF ( OK )
244: /* ANZEIGEN dieses Datensatzes                  */
245:               @ 12,00 CLEAR
246:               @ 12,23 SAY "Lieferanten Nr. : " +
                          TRIM(STR(LNUMMER,6,0))
247:               @ 14,23 SAY "Firma : "           +
                          TRIM(LFIRMA)
248:               @ 15,23 SAY "Name   : "          +
                          TRIM(LNAME)
249:               @ 16,23 SAY "TELEFON NUMMER : "  +
                          LTELEFON
250:               @ 19,00 SAY BILD1 + BILD1
251:               STORE 1 TO LAUF
252:               STORE ZAEHL + 1 TO ZAEHL
253:            ENDIF
254:            /**** 8 ****/
255:
256:         ELSE
257:
258:            /**** 9 ****/
259: /* FALL, bei dem nach der Lieferanten Nr.       */
260: /* und nach dem Namen des Lieferanten           */
261: /* gesucht werden soll.                         */
262:            INDEX LNUMMER
263:            STORE FIND INT(DSZUST) TO OK
264:            IF ( OK )
265:               IF ( LNAME = VLNAME )
266: /* ANZEIGEN dieses Datensatzes                  */
267:                  @ 12,00 CLEAR
268:                  @ 12,23 SAY "Lieferanten Nr. : "
                          TRIM(STR(LNUMMER,6,0))
269:                  @ 14,23 SAY "Firma : "
                          TRIM(LFIRMA)
270:                  @ 15,23 SAY "Name   : "
                          TRIM(LNAME)
271:                  @ 16,23 SAY "TELEFON NUMMER : "
                          LTELEFON
272:                  @ 19,00 SAY BILD1 + BILD1
273:                  STORE 1 TO LAUF
274:                  STORE ZAEHL + 1 TO ZAEHL
275:               ENDIF
276:            ENDIF
277:            /**** 9 ****/
278:         ENDIF
279:         IF ( LAUF = 1 )
280:            @ 21,13 SAY  " E = BEENDEN"
281:            @ 22,25 WAIT "WEITERMACHEN, BELIEBIGE
                      TASTE DRÜCKEN" TO EINGABE
282:         ENDIF
283: /* DATENSATZ ZEIGER um eins erhöhen             */
284:         STORE DSZU + 1 TO DSZU
285:
286: /*BEENDEN der Schleife für die Datensatz Anzeige*/
287:      ENDDO
288: /***********************************************/
289:    ENDIF
290:
291: /* FALL für die Anzeige der Listen am Bildschirm
   beenden.                                         */
292:  ENDIF
293: /* RÜCKKEHR zum Programm LSUCHE.TLK.            */
294:  @ 4,00 CLEAR
295:  @ 4,00 SAY BILD2 + BILD2
296:  @ 6,12 SAY "          SUCHEN neuer Lieferanten
                   ADRESSEN "
297:  @ 7,00 SAY BILD1 + BILD1
```

Listing 2: Das Programm LISUBI TLK.

```
 1:  /*                  LCOPY.TLK                    */
 2:  /**********************************************/
 3:  /*    Programmmodul zum Kopieren von            */
 4:  /*          Lieferanten Datensätzen.            */
 5:  /*Das Programm greift auf die Datenbank Beisp.zu*/
 6:  /*        copyright by Hans-Ulrich Mayer        */
 7:  /**********************************************/
 8:
 9:  /**** 1 ****/
10:  /* Abspeichern des gefundenen Datensatzes in
       lokale temporäre Variablen                    */
11:  /* Man befindet sich noch in der Datei ADRESSEN.*/
12:  STORE LNAME     TO TVLNAME
13:  STORE LNUMMER   TO TVLNR
14:  STORE LFIRMA    TO TVLFIRMA
15:  STORE LSTRASSE  TO TVLSTRASSE
16:  STORE LPLZ      TO TVLPLZ
17:  STORE LORT      TO TVLORT
18:  STORE LTELEFON  TO TVLTELEFON
19:  /**** 1 ****/
20:
21:  /**********************************************/
22:
23:  /**** 2 ****/
24:  /* Es wird in die temporäre Datei verzweigt,ohne*/
25:  /* daß die Datei ADRES.verlassen wird. Bei dem  */
26:  /* Verzweigen in die andere Datei muP der Datei */
27:  /* Name dem Zielschlüssel vorangestellt werden. */
28:  /* Verbindung mit einem Punkt                   */
29:  STORE (JUMP 0 TO ADRESSENCOPY.LNUMMER) TO TEMP
30:  /* Positionierung des ZEIGERS auf den letzten
       Datensatz.                                    */
31:  LAST
32:  /* Löschen des Datensatzinhaltes ( nur die Kopie
       im Arbeitsspeicher )                          */
33:  NEW
34:  /* EINTRAGEN der Merkmale in die Datei          */
35:  /* Der Dateiname muß dabei dem Merkmalsnamen
       vorangestellt werden.                         */
36:  REPLACE ADRESSENCOPY.LNUMMER  WITH TVLNR
37:  REPLACE ADRESSENCOPY.LNAME    WITH TVLNAME
38:  REPLACE ADRESSENCOPY.LFIRMA   WITH TVLFIRMA
39:  REPLACE ADRESSENCOPY.LSTRASSE WITH TVLSTRASSE
40:  REPLACE ADRESSENCOPY.LPLZ     WITH TVLPLZ
41:  REPLACE ADRESSENCOPY.LORT     WITH TVLORT
42:  REPLACE ADRESSENCOPY.LTELEFON WITH TVLTELEFON
43:  /**** 2 ****/
44:
45:  /**********************************************/
46:
47:  /**** 3 ****/
48:  /* Die Merkmale werden in der Datei abgespei.   */
49:  INSERT
50:  /* Rückkehr zu dem v. dem JUMP - Befehl
       aktuellen Datensatz                           */
51:  BACK
52:  /**** 3 ****/
```

Listing 3: Das Programm LCOPY TLK.

```
 1: /*              PROGRAMM LISUDR.TLK              */
 2: /***********************************************/
 3: /* Dies ist ein Beispiel für ein Programmodul zum*/
 4: /* Ausdrucken von Datens. auf dem Drucker. Sie   */
 5: /* können vorher mit der Druckerinitial. die     */
 6: /* spezifischen Steuercodes ihres Druckers, vorneh.
 7: /* Das Programm greift auf die Datenbank BEISP.zu.*/
 8: /*        copyright by Hans-Ulrich Mayer         */
 9: /***********************************************/
10:
11:  /**** 1 ****/
12:  /* FESTLEGUNG DER SCHALTERSTELLUNGEN            */
13:  /* Löschen des Bildschirmes                     */
14:  CLEAR
15:  /* Unterdrücken der automatischen Maskenanzeige */
16:  SET AUTOSC OFF
17:  /* Die Sonderregelungen für die Funktion FIND
       sind abgeschaltet                             */
18:  SET DBASE OFF
19: /*.Die Datenbank und die Datei wurde schon geöff */
20: /**** 1 ****/
21:
22: /************************************************/
23:
24:  /**** 2 ****/
25:  /* MENÜAUSGABE - Aufbau der Kopfzeile           */
26:  CLEAR
27:  /* aktuelle Datumsanzeige und der Uhrzeit       */
28:  @ 02     ,00  SAY " Datum "
29:  @ 03     ,00  SAY DATE()
30:  @ 02     ,72  SAY "Uhrzeit"
31:  @ 03     ,72  SAY TIME()
32:  @ 01     ,00  SAY BILD1 + BILD1
33:  @ ROW()  ,TAB SAY "           DATEN - SUCHE
        *** V 1.01 *** "
34:  @ ROW()+1,TAB SAY "flexible Anwendungspro-
                     grammierung mit ADIMENS Talk"
35:  @ ROW()+1,00  SAY BILD2 + BILD2
36:  @ ROW()  ,00  SAY BILD2 + BILD2
37:  @ ROW()+1,00  SAY " "
38:  @ ROW(),TAB SAY "       DRUCKEN von Datensätzen "
39:  @ ROW()+1, TAB SAY " "
40:  @ ROW()+1,00 SAY BILD1 + BILD1
41:  /**** 2 ****/
42:
43:  /**********************************************/
44:
45:  /**** 3 ****/
46:  /* LÖSCHEN des mittleren Bildschirmbereiches    */
47:  /* Durchlaufe die Schleife SOLANGE die Eingabe
        richtig ist.                                 */
48:  @ 10,00 CLEAR
49:  /* Lokale Variablendeklaration                  */
50:  STORE FALSE TO OK
51:  STORE SPACE(1) TO EINGABE
52:  STORE 0 TO ZAEHL
53:  STORE "F05" TO MSG
54: /**** 3 ****/
55:
56: /***********************************************/
57:
58: /**** 4 ****/
59: /* Dieser Programmabschnitt kopiert die gefund. */
60: /* Datensätze in eine TEMPORÄRE DATEI.
     Die temporäre                                  */
61: /* Datei dient vor allem der Indizierung der
     Daten.                                         */
62: /* Somit hat man die Möglichkeit einer mehrmaligen*/
63: /* Ausgabe.                                     */
64:  @ 10,02 SAY "Die gefundenen Datensätze werden
                alle in eine"
65:  @ 11,02 SAY "        temporäre Datei kopiert. "
66:  @ 13,02 SAY "Anzahl der zu kopierenden Datensätze:
               " +TRIM(STR(GD,6,0))
67:  @ 14,02 SAY "schon kopierte Datensätze          :
               " +TRIM(STR(ZAEHL,6,0))
68:  @ 18,02 SAY "Kopierstart BITTE  WARTEN !!! "
69:  @ 15,00 SAY BILD1 + BILD1
70:  @ 17,00 SAY BILD1 + BILD1
71:  @ 19,00 SAY BILD1 + BILD1
72:  @ 10,54 SAY "      DRUCKEN "
73:  @ 12,54 SAY "Bitte Papier einlegen und "
74:  @ 13,54 SAY "Drucker eingeschalten ??? "
75:  /**** 4 ****/
76:
77:  /**********************************************/
78:
79:  IF ( DSZUST = "0" )
80:
81:     /**** 5 ****/
82:     STORE ( JUMP VLNAME TO LNAME ) TO OK
83:
84:    /*******************************************/
85:    /* SOLANGE die Zählvariable < Anzahl der
         gefundenen Datensätze GD      */
86:     DO WHILE NOT (ZAEHL = GD)
87:     /******************************************/
88:
89:        @ 14,42 SAY TRIM(STR(ZAEHL+1,6,0))
90:        /**************************************/
91:        /* SPRUNG ins Untermodul LCOPY.TLK    */
92:        /* Die gefundenen Datensätze werden in die*/
93:        /* temporäre Datei kopiert            */
94:        DO "LCOPY"
```

Listing geht weiter...

```
 95: /***********************************************/
 96:         STORE ZAEHL + 1 TO ZAEHL
 97:         SKIP 1
 98:     ENDDO
 99:
100: /* ZEIGER auf den Anfang der Datei zurücksetzen.*/
101:     BACK
102:     /**** 5 ****/
103:
104:  ELSE
105:
106:     /**** 6 ****/
107:     /*******************************************/
108:     /* SOLANGE die Zählvariable < gefundenen
           Datensätze GD                     */
109:     DO WHILE NOT ( ZAEHL = GD )
110:     /*******************************************/
111:
112:        @ 14,42 SAY TRIM(STR(ZAEHL+1,6,0))
113:        STORE STR(DSZU,6,0) TO DSZUST
114:        IF ( VLNAME = "" )
115:           INDEX LNUMMER
116:           STORE FIND INT(DSZUST) TO OK
117:           IF ( OK )
118:     /*******************************************/
119:     /* SPRUNG ins Untermodul LCOPY.TLK          */
120:     /* Die gefundenen Datensätze werden in      */
121:     /* die temporäre Datei kopiert.             */
122:               DO "LCOPY"
123:     /*******************************************/
124:               STORE ZAEHL + 1 TO ZAEHL
125:           ENDIF
126:        ELSE
127:           INDEX LNUMMER
128:           STORE FIND INT(DSZUST) TO OK
129:           IF (OK)
130:              IF (LNAME = VLNAME)
131:     /*******************************************/
132:     /* SPRUNG ins Untermodul LCOPY.TLK          */
133:     /* Die gefundenen Datensätze werden in      */
134:     /*  die temporäre Datei kopiert.            */
135:                 DO "LCOPY"
136:                 /*******************************/
137:                 STORE ZAEHL + 1 TO ZAEHL
138:              ENDIF
139:           ENDIF
140:        ENDIF
141:        STORE DSZU + 1 TO DSZU
142:     ENDDO
143:  ENDIF
144:  /**** 6 ****/
145:
146:  /*********************************************/
147:
148:  /**** 7 ****/
149:  @ 18,54 SAY "                Kopierende "
150:  @ 22,20 SAY "Bitte eine beliebige Taste drücken "
151:  WAIT
152:  /* Umschalten auf die temporäre Datei
        ADRESSENCOPY    */
153:  USE ADRESSENCOPY
154:  /**** 7 ****/
155:
156:  /*********************************************/
157:
158:  /**** 8 ****/
159:  /*********************************************/
160:  /* SOLANGE die Eingabe nicht richtig ist.
        Plausibilitätskontrolle          */
161:  DO WHILE NOT (UPPER (EINGABE) = "N" OR UPPER
     (EINGABE) ="L")
162:  /*********************************************/
163:
164:      STORE "N" TO EINGABE
165:      @ 10,00 CLEAR
166:      /* ABFRAGE, ob nach dem Namen oder der L.-Nr.
           sortiert werden soll.   */
167:      @ 10,14 SAY "Bitte entscheiden Sie sich für
                   einen Sortierschlüssel"
168:      @ 13,14 SAY "sortieren nach dem LIEFERANTEN
                   NAMEN :  -> 'N' drücken"
169:      @ 14,14 SAY "sortieren nach der LIEFERANTEN
                   NUMMER: -> 'L' drücken"
170:      @ 11,14 SAY "es wird dabei jeweils aufsteigend
                   sortiert. "
171:      @ 15,33 SAY "EINGABE: " GET EINGABE
```

```
172:      @ 16,00 SAY BILD1 + BILD1
173:      READ
174:      /* FALLUNTERSCHEIDUNG mit einer
            Plausibilitätskontrolle.                 */
175:      DO CASE
176:         CASE (UPPER (EINGABE) = "N" )
177:            STORE 1 TO SORT
178:         CASE (UPPER (EINGABE) = "L" )
179:            STORE 2 TO SORT
180:         OTHERWISE
181:            BELL
182:            @ 19,00 SAY BILD1 + BILD1
183:            @ 20,20 SAY "FALSCHE EINGABE !!! BITTE
                     WIEDERHOLEN "
184:            @ 21,00 SAY BILD1 + BILD1
185:            @ 22,20 SAY  " beliebige Taste drücken "
186:            WAIT
187:      ENDCASE
188:   ENDDO
189:
190:   STORE " " TO EINGABE
191:   /**** 8 ****/
192:
193:   /*********************************************/
194:
195:   /**** 9 ****/
196:   /*********************************************/
197:   /* SOLANGE, die Eingabe nicht richtig ist.
         Plausibiltätskontrolle.        */
198:   DO WHILE NOT(UPPER(EINGABE) = "A" ORUPPER(EINGABE)
                                 = "T")
199:   /*********************************************/
200:
201:      STORE " " TO EINGABE
202:      @ 10,00 CLEAR
203:      /* ABFRAGE, ob eine Adressen- oder eine
            Telefon-Liste erstellt wird             */
204:      @ 10,14 SAY "Bitte entscheiden Sie sich für
                     eine Liste "
205:      @ 12,14 SAY "ADRESSEN Liste der Lieferanten :
                  -> 'A' drücken "
206:      @ 13,14 SAY "TELEFON  Liste der Lieferanten :
                  -> 'T' drücken "
207:      @ 15,33 SAY "EINGABE: " GET EINGABE
208:      @ 16,00 SAY BILD1 + BILD1
209:      READ
210:      @ 17,27 SAY "Ist Ihr Drucker bereit ? "
211:/* Fallunterscheidung mit Plausibilitätskontrolle*/
212:      DO CASE
213:    /* Adressen Liste ausdrucken lassen          */
214:         CASE (UPPER (EINGABE) = "A")
215:            STORE " " TO EINGABE
216:
217:    /*********************************************/
218:    /* SOLANGE die Eingabe nicht richtig ist.
          Plausibiltät !                            */
219:            DO WHILE NOT (UPPER (EINGABE) = "D" OR
                UPPER (EINGABE) = "A")
220:    /*********************************************/
221:
222:               STORE "D" TO EINGABE
223:               @ 18,00 SAY BILD1 + BILD1
224:               @ 19,10 SAY "DRUCK START -> 'D'
                          drücken "
225:               @ 19,50 SAY "ABBRECHEN   -> 'A'
                     drücken "
226:               @ 20,00 SAY BILD1 + BILD1
227:               @ 21,33 SAY "EINGABE :" GET EINGABE
228:               READ
229:    /*********************************************/
230:    /* SPRUNG ins Untermodul LDRUCKAD.TLK.       */
231:    /* Dort wurde das Format des Ausdruckes      */
232:    /* der ADRESSENLISTE festgelegt.             */
233:               DO "LDRUCKAD"
234:    /*********************************************/
235:
236:            ENDDO
237:
238:  /* FALL : TELEFON LISTE auf dem Drucker
        ausdrucken                                   */
239:         CASE (UPPER (EINGABE) = "T" )
240:            STORE " " TO EINGABE
241:
242:      /*******************************************/
243:      /* SOLANGE, die Eingabe nicht richtig ist. */
```

Listing geht weiter...

```
244:            DO WHILE NOT (UPPER(EINGABE )= "D" OR
                UPPER(EINGABE) = "A")
245:      /******************************************/
246:
247:               STORE "D" TO EINGABE
248:               @ 18,00 SAY BILD1 + BILD1
249:               @ 19,10 SAY " DRUCK START ->'D'
                               drücken"
250:               @ 19,50 SAY " ABBRECHEN   ->'A'
                               drücken"
251:               @ 20,00 SAY BILD1 + BILD1
252:               @ 21,33 SAY "EINGABE :" GET EINGABE
253:               READ
254:       /*****************************************/
255:       /* SPRUNG ins Untermodul LDRUCKTE.TLK.     */
256:       /* Dort wurde das Format des Ausdruckes    */
257:       /* der TELEFON LISTE festgelegt.           */
258:               DO "LEER"
259:       /*****************************************/
260:
261:            ENDDO
262:
263:      /* ANDERENFALLS :  Fehlermeldung und Warnton
            erzeugen.                               */
264:        OTHERWISE
265:            BELL
266:            @ 18,00 CLEAR
267:            @ 19,00 SAY BILD1 + BILD1
268:            @ 20,20 SAY "FALSCHE EINGABE !!! BITTE
                    WIEDERHOLEN "
269:            @ 21,00 SAY BILD1 + BILD1
270:            @ 22,20 SAY "        beliebige Taste
                        drücken "
271:            WAIT
272:       ENDCASE
273:
274:    ENDDO
275:    /**** 9 ****/
276:
```

Listing 4:Das Programm LISUDR.TLK

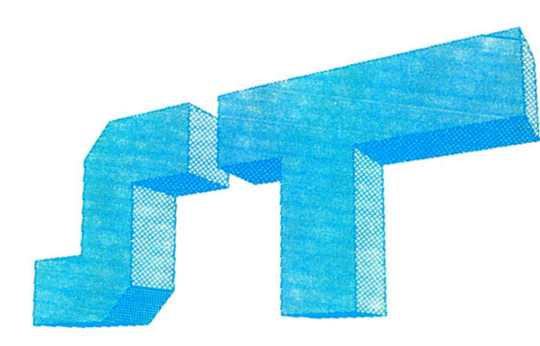
```
  1:    /*                 LDRUCKAD.TLK               */
  2:    /*******************************************/
  3:
  4:    /**** 1 ****/
  5:    /* Datensatzzeiger auf den Anfang positionieren*/
  6:    USE ADRESSENCOPY
  7:    IF (SORT = 1)
  8:       INDEX LNAME
  9:    ELSE
 10:       INDEX LNUMMER
 11:    ENDIF
 12:    FIRST
 13:    /* LOKALE Variablendeklaration              */
 14:    STORE 1 TO GDS
 15:    STORE 1 TO SEITE
 16:    COUNT ALL TO GDS
 17:    FIRST
 18:    CLEAR
 19:    /**** 1 ****/
 20:
 21:  /*******************************************/
 22:
 23:   DO CASE
 24:      CASE ( UPPER(EINGABE) = "D" )
 25:         /**** 2 ****/
 26:         SET CONSOLE OFF
 27:         SET PRINTER ON
 28:         /* DRUCKEN der Kopfzeile                */
 29:         ? "               ADRESSEN LISTE vom : "
 30:         ?? DATE()
 54:               /* Einzelblatt auswerfen */
 55:               ? CHR(27)+CHR(25)+CHR(82)
 56:               SET PRINTER OFF
 57:               SET CONSOLE ON
 58:               STORE SEITE+1 TO SEITE
 59: /* Abfrage, ob eine neue Seite eingelegt wurde ?*/
 60:               @ 0,0 CLEAR
 61:               @ 10,20 SAY "NEUE SEITE
                                 WEITERDRUCKEN ?"
 62:               @ 12,20 WAIT"             (J/N)        "
                                               TO EINGABE
 63:            DO CASE
 64:               CASE ( UPPER ( EINGABE) = "J" )
 65:                  STORE " " TO EINGABE
 66:                  SET PRINTER ON
 67:                  SET CONSOLE OFF
 68:   /* DRUCKEN der Kopfzeile                        */
 69:                  ? "         ADRESSEN LISTE vom : "
 70:                  ?? DATE()
 71:                  ??"        SEITE: "
 72:                  ?? TRIM(STR(SEITE,3,0))
 73:                  ?  BILD1 + BILD1
 74:                  ?  "NR      FIRMA          STRASSE"
 75:                  ?? "                 PLZ  WOHNORT "
 76:                  ?  BILD1 + BILD1
 77:                  STORE 6 TO Z
 78:               CASE ( UPPER(EINGABE)="N" )
 79:                  STORE "E" TO EINGABE
 80:               OTHERWISE
 81:                  BELL
 82:                  @ 18,00 CLEAR
 83:                  @ 19,00 SAY BILD1 + BILD1
 84:                  @ 20,20 SAY "FALSCHE EINGABE !!!
                            BITTE WIEDERHOLEN"
 85:                  @ 21,00 SAY BILD1 + BILD1
 86:            ENDCASE
 87:            SET PRINTER ON
 88:            SET CONSOLE OFF
 89:         ENDIF
 90:
 91:      ENDDO
 92:
 93:   /* Auswurf eines einzelnen Papierblattes.      */
 94:      ? CHR(27)+CHR(25)+CHR(82)
 95:      SET PRINTER OFF
 96:      SET CONSOLE ON
 97:      /**** 2 ****/
 98:
 99:   CASE (UPPER(EINGABE) = "A" )
100:
101:      /**** 3 ****/
102:      /* NICHTS  TUN */
103:      /**** 3 ****/
104:
105:   OTHERWISE
106:
107:      /**** 4 ****/
108:      BELL
109:      @ 18,00 CLEAR
110:      @ 19,00 SAY BILD1 + BILD1
111:      @ 20,20 SAY "FALSCHE EINGABE !!! BITTE
                    WIEDERHOLEN "
112:      @ 21,00 SAY BILD1 + BILD1
113:      WAIT
114:      /**** 4 ****/
115:
116:  ENDCASE
117:
118:  /**********************************************/
119:
120:  /**** 5 ****/
121:  /* NEUER BILDSCHIRM AUFBAU                      */
122:  /* MENÜAUSGABE - Aufbau der Kopfzeile           /
123:  CLEAR
124:  /* aktuelle Datumsanzeige und der Uhrzeit       */
125:  @ 02     ,00  SAY " Datum "
126:  @ 03     ,00  SAY DATE()
127:  @ 02     ,72  SAY "Uhrzeit"
128:  @ 03     ,72  SAY TIME()
129:  @ 01     ,00  SAY BILD1 + BILD1
130:  @ ROW()  ,TAB SAY "            DATEN - SUCHE
                                 *** V 1.01 *** "
131:  @ ROW()+1,TAB SAY "flexible
    Anwendungsprogrammierung mit ADIMENS Talk"
132:  @ ROW()+1,00  SAY BILD2 + BILD2
133:  @ ROW()  ,00  SAY BILD2 + BILD2
134:  @ ROW()+1,00  SAY " "
135:  @ ROW(),TAB SAY "     DRUCKEN von Datensätzen "
136:  @ ROW()+1, TAB SAY " "
137:  @ ROW()+1,00 SAY BILD1 + BILD1
138:  /**** 5 ****/
139:
140:  /**********************************************/
```

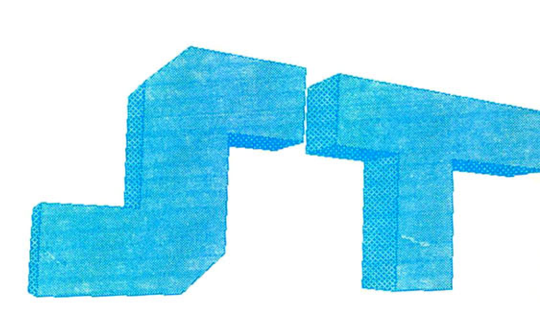
Listing 5: Das Programm LDRUCKAD. TLK

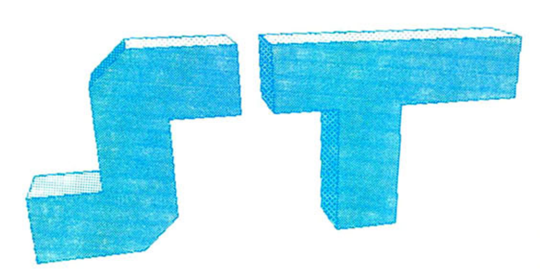
SCHNELLE 3D AUF DEM ST

Flächendeckende Objektgrafik in GFA-BASIC

Teil II







Im vorangegangenen Artikel, der zusammen mit dem Programmlisting als einzelner Artikel gerade noch vertretbar schien, wurde ein Programm zur Erstellung und Darstellung dreidimensionaler Objektgrafik vorgestellt. Einige Fragen, die noch offenbleiben mußten sowie einige hilfreiche Funktionen zur Programmerweiterung werden jetzt hier behandelt.

GFA-Freaks werden es schon längst erkannt haben, die Datenstruktur des Felds *Xyz%(2,Pun_anz)* für die Punkte weicht von der des Flächenfelds *Fl% (Eck_max+3,Fl_max)* ab. So werden die Punkte von *1* bis *P_anz* durchnumeriert im Gegensatz zu den Flächen, die von *0* bis *F_anz-1* zählen. Die Einträge im Punktfeld entsprechen den Koordinaten *X,Y,Z* des jeweiligen Punkts. Die Einträge im Flächenfeld von *0* bis *Eck_max* sind die Nummern der Punkte analog zum Feld *Xyz%()* in der 'Korkenzieherreihenfolge'. Die drei Einträge, die darauf folgen *(Eck_max+1 bis Eck_max+3)*, sind zuerst die Anzahl der Punkte - 1 und dann 2 Werte für die Farb- oder Musterinformation.

Mit Kenntnis dieser Struktur erschließt sich einem die hier neu vorgestellte Prozedur *F_use* recht gut. Am linken Rand des Bildschirms werden alle vorhandenen Flächen aufgelistet. Sollten es mehr sein als der Monitor anzeigen kann, läßt sich mit Pfeilen nach oben oder nach unten scrollen. Jeder Kasten zeigt Füllfarbe oder -muster, die Flächennummer und die Anzahl der Punkte dieser Fläche. Eine Mausbewegung auf einen der Kästchen, und eine Inrandung (neudeutsch: Gegensatz von Umrandung) erscheint. Zudem blinken die Punkte dieser Fläche im Anzeigefeld, falls sie sichtbar sind. Ein Mausklick auf das Flächenkästchen, und es erscheint eine Alarmbox mit den Möglichkeiten, Füllmuster oder -farbe abzuändern oder die Fläche wegzuwerfen. So lassen sich nun falsch konstruierte Flächen leicht entfernen, und die Umwandlung von Objekten verschiedener Auflösungsstufen ist auch schnell gemacht.

Diskettenkomfort

Geradezu notwendigen Komfort bringen die neuen Menüpunkte "OBJ LADEN" und "SPEICHERN". Nomen est Omen, es sollte nur erwähnt werden, daß mit Rechtsklick auf diese Menüpunkte ein Pfadname vorgegeben werden kann, und die Dateiart **.3D?* sollte am besten à la DEGAS für die unterschiedlichen Auflösungsarten verwendet werden (3D1 nied-

rige Auflösung, 3D2 mittel, 3D3 hochauflösend).

Die Dateien werden nach einem einfachen Verfahren zusammengequetscht, man muß jedoch darauf achten, daß die Variable *Eck_max* nicht kleiner gewählt wird als die maximale Eckenzahl des zu ladenden Objekts, sonst erfolgt eine Warnung vom Programm. In diesem Zusammenhang der Hinweis, daß die Variablen *Pun_max* (maximale Punktanzahl), *Fl_max* (maximale Flächenanzahl) und *Eck_max* (maximale Anzahl der Eckpunkte einer Fläche) je nach Belieben verändert werden können.

Unbekannte Tiefen

Eine neue Funktion ist auch "TIEFENVERZERRUNG". Hiermit können die Grundwerte von *Darstx* und *Darsty* verändert werden, die die Anzeige der Tiefe im Eingabefenster regeln. Dies ist vor allem dann vonnöten, wenn mehrere Punkte so dicht beieinanderliegen, daß die Flächenerstellung zur Tortur wird. Nach einem Klick auf "TIEFEN-VERZERRUNG" verändert sich die Darstellung abhängig von der Mausposition. Sie sollte langsam von der oberen rechten Ecke des ersten Quadranten Richtung Ursprung geändert werden. Je näher man dem Mittelpunkt kommt, desto stärker die Verzerrung. Bei der Position, an der die Punktauflösung günstig ist, genügt ein Mausklick, und die neue Verzerrung wird von nun an vom Programm verwendet.

Kaum erwähnenswert ist die Programmabbruchsfunktion, die einem die Fingerakrobatik von <Alternate-Shift-Control> erspart. Bei Programmversionen, die man kompilieren will, sollte "End" durch "Quit" ersetzt werden.

Eine weitere Änderung ergibt sich in der Angabe der *X*-,*Y*-,*Z*-Werte. Die *X*- und *Y*-Koodinate können jetzt genau wie der *Z*-Wert mit Mausdruck oder -klick verändert werden. Damit verschiebt sich das Koordinatenkreuz, und Objekte, die größer sind als das Eingabefenster, können bearbeitet werden.

Die letzte Veränderung bezieht sich auf die neue Menüfunktion "QUADER ERSTELLEN". Hiermit kann ein beliebig großer Quader an jeder Stelle parallel zu den Koordinatenachsen erstellt werden. Nach Klick auf die Menüfunktion muß man sich zuerst entscheiden, ob man den Quader anhand des Mittelpunkts oder des vorderen linken unteren Eckpunkts bestimmen will. Danach können die Größenangaben für drei Seiten angegeben werden, und dann erfolgt die Eingabe des Eckpunkts bzw. des Mittelpunkts. Nun brauchen nur noch die 6 Muster oder Farben für die Flächen des Quaders eingegeben zu werden, und er erscheint in dem Anzeigefeld.

Versteckte Hilfe

In der bisherigen Programmbeschreibung wurden zwei Funktionen über die das Programm verfügt, nicht erläutert. Die eine bezieht sich auf das selbsttätige Überprüfen der 'Korkenzieherregel' bei der Flächenerstellung. Bei allen exakt definierten ausgefüllten Objekten gehört jede Linie zu genau 2 Flächen. Und bei konsequenter Anwendung der 'Korkenzieherregel' stellt sich heraus, daß die Flächenecken jede Linie immer gegenläufig beschreiben. Abbildung 1 veranschaulicht diesen verbal schwer zu beschreibenden Effekt. Hier setzt auch das Programm ein und überprüft in der Prozedur *F_akt*, ob nach Fertigstellung einer Fläche diese mit irgendeiner Linie einer anderen Fläche dieselbe Richtung beschreibt. Falls ja, werden die Eckpunkte in umgekehrter Reihenfolge abgespeichert (unter Zuhilfenahme des *X%()*-Feldes). Dies erklärt, warum neue Flächen immer an bereits existierende angrenzen sollten, denn nur dann kann das Programm die richtige 'Korkenzieherorientierung' überprüfen bzw. einsetzen.

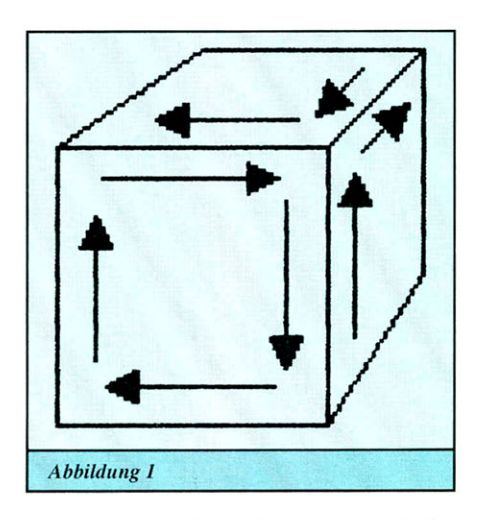


Abbildung 1

Ein anderer Aspekt ist die Darstellung des Objekts mit den den Linienmustern innewohnenden Angaben über Flächenzugehörigkeit. Dieser für die Erstellung vollständiger Objekte hilfreiche Modus, der sich mit "INFO + ANZEIGE" umschalten läßt, wird durch die Überprüfung in der Prozedur *"Show"* (If Grmbo...) erreicht. Hier wird jede Linie daraufhin überprüft, ob sie bei einer anderen Fläche auch schon vorhanden ist und, falls ja, gestrichelt dargestellt. Diese Überprüfung geht (Brute Force) alle Flächen und deren Linien durch und bricht nur bei einer zweiten Linie ab. Dieser Aufwand, der für jede einzelne Linie aufgebracht werden muß, erklärt den langsamen Bildaufbau in dieser Darstellungsart. Programmtechnisch kann hier noch viel verbessert werden, da jede gestrichelte Linie ja mindestens zweimal berechnet und ausgegeben wird. Um das Programm nicht noch länger und unübersichtlich zu machen, mußte hierauf leider verzichtet werden.

Ein Hinweis noch für die Erstellung von großen Flächen mit vielen Eckpunkten. Hier muß darauf geachtet werden, mit den ersten drei Punkten 'Korkenzieherrichtigkeit' zu erzielen, denn nur diese werden für die Berechnung des Kreuzprodukts ausgewertet. Danach entgegengesetzte Laufrichtung beeinflußt die Darstellung nicht im mindesten.

Die Einbindung der neuen Prozeduren dürfte nicht schwerfallen. Außer den DATA-Zeilen muß nur noch der Befehlsaufruf in der Hauptschleife ersetzt werden (siehe Listing). Die Prozeduren selbst können einfach an das bisherige Programm angehängt werden. Und nach all der Mühe bleibt lediglich noch der Wunsch, daß sie mit beeindruckenden 3D-Konstruktionen und erfolgreichen Programmerweiterungen gekrönt werden. Die Redaktion würde sich freuen, an Ihren Erfahrungen und Ergebnissen teilzuhaben.

Martin A. Wielebinski

Literatur:

William M. Newman/Robert F. Sproull, Grundzüge der interaktiven Computergrafik: 1986, McGraw-Hill, Hamburg.

c't 1984, Heise Verlag, Hannover. (Sehr unterhaltsame Artikelserie über ein 6502-Assemblerpaket für dreidimensionale Drahtmodelle.)

c't S. 144 ff Von Vektoren und Volumina: Nr. 2 1988, Heise Verlag, Hannover.

CHIP SPECIAL Computergrafik: März 1981, Vogel Verlag, Würzburg.

GFA-DRAFT plus für ST

das leistungsfähige CAD-Programm für alle, die entwerfen, konstruieren und zeichnen.



- Voll GEM-gesteuertes leistungsfähiges CAD-Programm
- Maßstabgerechtes Erstellen von Zeichnungen in Zoll und mm
- Bildausschnitte und Symbole beliebig manipulierbar und gradweise drehbar
- Mächtige Zeichenfunktionen wie z. B. Lot, Winkel zu Geraden, Kreis durch 3 Punkte, etc.
- 255 Zeichenebenen je bis DIN A 0 (10 gleichzeitig darstellbar)
- Schraffieren und Bemaßen von beliebigen Flächen
- Eingebaute Kommandosprache
- Einfachste Handhabung
- Läuft auf fast allen Plottern und Druckern
- Symbole und Bibliotheken in beliebiger Menge anlegbar
- GEM-Desktop ist im Lieferumfang enthalten

DM 349,–

Symbolbibliotheken
Elektrotechnik
Pneumatik/Hydraulik
Maschinenbau/Verbindungselemente
Möbel
je DM 149,–

DIN Fachbericht 4
DM 249,50

GFA-CLUB
GFA-ST/PC-Software
bitte Info anfordern
...Anruf genügt.

GFA Systemtechnik GmbH
Heerdter Sandberg 30-32
D-4000 Düsseldorf 11
Telefon 0211/5504-0



```
1:   '
2:   ' ***********************************************
3:   ' *                                             *
4:   ' * 3D-Grafik Edier- und Demoprogramm für die ST
       von Martin A. Wielebinski.                    *
5:   ' * Teil 2, die bisher 'reservierten' Funktionen;
       Flächeneditor, Load & Save.                   *
6:   ' *
                                                     *
7:   ' ***********************************************
8:   '
9:   ' Um die zusätzlichen Prozeduren lauffähig zu
       machen müssen zwei kleine
10:  ' Änderungen des Originalprogramms vorgenommen
       werden.
11:  '
12:  ' Die Befehlsaufrufe in der Hauptschleife lauten
       nunmehr:
13:  '
14:  If Com<10
15:    On Com Gosub Wert,Wert,Wert,In_gr,Verz,Gitt,Phol,
       Fhol,G3d
16:  Else
17:    On Com-9 Gosub F_use,Quader,D,D,D,D,D,D,Olad,Osp,
       Pstp
18:  Endif
19:  '
20:  ' Und die Data-Zeilen mit den Menütexten
       folgendermaPen:
21:  '
22:  Data "- X:                +"
23:  Data "- Y:                +"
24:  Data "- Z:                +"
25:  Data INFO + ANZEIGEMODUS
26:  Data TIEFEN-VERZERRUNG
27:  Data GITTER | SETZEN
28:  Data PUNKTMOD|XYZ-EING
29:  Data FLÄCHEN ERSTELLEN
30:  Data 3D-GRAFIK STARTEN
31:  Data FLÄCHEN EDITIEREN
32:  Data QUADER EINGEBEN
33:  Data FREI,FREI,FREI
34:  Data FREI,FREI,FREI
35:  Data OBJ LADEN|PFAD BEST
36:  Data SPEICHERN|PFAD BEST
37:  Data 3D PROGRAMM BEENDEN
38:  '
39:  ' Die nun folgenden Prozeduren werden einfach an
       das Originalprogramm
40:  ' angehängt, eigene Funktionen müPten problemlos
41:  ' weiterbenützt werden können.
42:  '
43:  Procedure Verz   ! Bildschirmverzerrung einstellen
44:    Local I,J,A$
45:    A$="Darstellungsverzerrung:|Je näher sie die Maus
         dem|Mittelpunkt bringen"
46:    Gosub Ali(0,A$+" desto|stärker die Verzerrung.",
           1,"Aha!")
47:    Gosub Mclr
48:    J=Grmbo          ! Schnelle Darstellung
49:    Grmbo=0
50:    Repeat
51:      Darstx=(Mousex-Ymi)/10    ! Verzerrung x und y
                                     anhand
52:      Darsty=(Ymi-Mousey)/10    ! des Mittelpunkts
                                     berechnen
53:      If Darstx=0   ! Dürfen aber nicht 0 betragen
54:        Darstx=1
55:      Endif
56:      If Darsty=0   ! dito
57:        Darsty=1
58:      Endif
59:      Gosub Show    ! Veränderungen anzeigen
60:    Until Mousek    ! bis Maustaste geklickt wird
61:    Grmbo=J         ! Originaldarstellung wieder
                         herstellen
62:  Return
63:  '
64:  Procedure Hol_pfad  ! Kein Kommentar
65:    Gosub Inbox("DEN ZUGRIFFSPFAD FÜR|DIE DATEN
                 ANGEBEN.|zB: A:\GFA\GRAFIK\3D\",1)
66:    Disk$=Wert$
67:  Return
68:  '
69:  Procedure Osp          ! Objekt abspeichern
70:    Local I,J,Maxeck
71:    If Mk=2
72:      Gosub Hol_pfad     ! Bei Rechtsklick Pfad
                              holen
73:    Else                 ! sonst:
74:      Fileselect Disk$+"*.3D?","",F$
75:      If F$<>""
76:        Open "O",#1,F$    ! Falls guter Dateiname
                               öffnen
77:        Maxeck=0
78:        For I=0 To F_anz-1
79:          If Fl%(Eck_max+1,I)>Maxeck
80:            Maxeck=Fl%(Eck_max+1,I)    ! Maximale
                       Eckenzahl einer Fläche des
81:          Endif          ! Objekts ermitteln
82:        Next I
83:        Gosub Byte(P_anz) ! Punktanzahl,
84:        Gosub Byte(F_anz) ! Flächenanzahl
85:        Gosub Byte(Maxeck)! und maximale Eckenanzahl
                               zur Datei
86:        If P_anz
87:          For I=1 To P_anz
88:            For J=0 To 2
89:              Gosub Byte(Xyz%(J,I)) ! Alle Punkt-
                                         koordinaten,
90:            Next J
91:          Next I
92:        Endif
93:        If F_anz
94:          For I=0 To F_anz-1
95:            For J=0 To Maxeck
96:              Gosub Byte(Fl%(J,I))! alle relevanten
                                    Flächennummern
97:            Next J
98:            For J=Eck_max+1 To Eck_max+3
99:              Gosub Byte(Fl%(J,I))! und die
                    Flächeninformationen ausgeben.
100:           Next J
101:         Next I
102:       Endif
103:       Close #1             ! Das war es schon...
104:     Endif
105:   Endif
106: Return
107: '
108: Procedure Byte(W)         ! Diese Proz. wirft 2
                                 Bytes zur Diskette
109:   W=W+32768
110:   Out #1,(W Div 256)
111:   Out #1,(W Mod 256)
112: Return
113: '
114: Procedure Olad            ! Lädt ein Objekt von Disk
115:   Local I,J,Maxeck
116:   If Mk=2
117:     Gosub Hol_pfad        ! Rechtsklick -> holt Pfad
118:   Else
119:     Fileselect Disk$+"*.3D?","",F$
120:     If F$<>""
121:       Open "I",#1,F$      ! Falls guter Dateiname
                                 anfügen
122:       Gosub Wrd
123:       I=Ww                ! und 3 Words einlesen
124:       Gosub Wrd
125:       J=Ww
126:       Gosub Wrd
127:       Maxeck=Ww
128:       If Maxeck>Eck_max   ! Maximale Eckenzahl
                                 möglich?
129:         Gosub Ali(3,"Objekt hat mehr Ecken|als
                 die Var. Eck_max.",1,"Stop")
130:       Else
131:         P_anz=I             ! JA, Werte übernehmen
132:         F_anz=J
133:         If P_anz
134:           For I=1 To P_anz
135:             For J=0 To 2
136:               Gosub Wrd
137:               Xyz%(J,I)=Ww ! Alle Punktkoordinaten
138:             Next J
139:           Next I
140:         Endif
141:         If F_anz
142:           For I=0 To F_anz-1
143:             For J=0 To Maxeck
144:               Gosub Wrd
```

Listing geht weiter...

```
145:            Fl%(J,I)=Ww    ! alle Flächennummern
146:          Next J
147:          For J=Eck_max+1 To Eck_max+3
148:            Gosub Wrd
149:            Fl%(J,I)=Ww ! und alle Flächeninfos
                                  lesen.
150:          Next J
151:        Next I
152:      Endif
153:    Endif
154:    Close #1                ! das wars...
155:  Endif
156: Endif
157: Return
158: '
159: Procedure Wrd      ! Diese Procedure liest 2 Bytes
160:   W1=Inp(#1)
161:   Ww=W1*256+Inp(#1)-32768
162: Return
163: '
164: Procedure Pstp     ! Programm abbrechen???
165:   Local A$
166:   A$="Programm wirklich|beenden und alle|Daten
       vernichten?"
167:   Gosub Ali(3,A$,2," JA |Nein")
168:   If But=1
169:     Cls               ! Jawoll!!!
170:     End
171:   Endif
172: Return
173: '
174: ' Diese umfangreiche Procedure ermöglicht die
     Edierung von Flächen:
175: '
176: Procedure F_use
177:   Local Mmx,Mmy,Mmk,I,Stnr,Aktnr,Rnr,A$
178:   If F_anz
179:     A$="Sie können links die Fläche|auswählen und
         durch|Mausklick edieren.|"
180:     Gosub Ali(0,A$+"Zurück mit Rechtsklick!",1,
            "Aha!")
181:     Deffill 0,1
182:     Pbox 0,0,50,Ysl
183:     Text 10,Tyo,"^^^^^"
184:     Text 10,Yty*19+Tyo,"vvvvv" ! Bildschirm
                                   ruinieren
185:     Color 1
186:     Stnr=0                    ! Startnummer = 0
187:     '
188:     Starty:
189:     Deffill 0,1
190:     Pbox 0,Yty,50,Yty*19       ! Flächenbox am
                                   linken Schirmrand
191:     Aktnr=-1
192:     For I=1 To 18
193:       Box 0,Yty*I,50,Yty*(I+1) ! Box ausgeben
194:       If Stnr+I-1<F_anz
195:         A$=Str$(Stnr+I)+" "+Str$(Fl%(Eck_max+1,
               Stnr+I-1)+1)+"P"
196:         Text 28-3*Len(A$),Yty*I+Tyo,A$ ! Text
                                             ausgeben
197:         Mmy=Fl%(Eck_max+2,Stnr+I-1)
198:         Mmx=Fl%(Eck_max+3,Stnr+I-1)
199:         If Mmx
200:           Deffill 1,Mmx,Mmy
201:         Else
202:           Deffill Mmy,1
203:         Endif
204:         Pbox 0,Yty*I,8,Yty*(I+1) ! Füllfarbe oder
                                        -muster
205:       Endif
206:     Next I
207:     Do                             ! Warteschleife
208:       Mouse Mmx,Mmy,Mmk
209:       If Mmx<40                    ! Maus in Box?
210:         I=Mmy Div Yty
211:         If I>0 And I<19 And Stnr+I-1<F_anz  ! ja,
                                gültige Flächennummer?
212:           If I<>Aktnr
213:             Graphmode 3  ! ja, aktivieren
214:             If Aktnr<>I  ! alte ausschalten???
215:               Box 10,Yty*Aktnr+2,48,Yty*(Aktnr+1)
                   -2! jawoll
216:             Endif
217:             Box 10,Yty*I+2,48,Yty*(I+1)-2 ! In-
                                             randung
218:             Aktnr=I              ! aktive Position
219:             Rnr=Stnr+Aktnr-1   ! aktuelle
                                   Flächennummer
220:             Graphmode 0
221:           Endif
222:         Endif
223:         If Mmk=1          ! Maus auch noch gedrückt?
224:           If I>0 And I<19  ! ja, auf
                                 Flächennummer?
225:             If Aktnr<>-1
226:               Gosub Ali(2,"Fläche löschen oder|
                neu einfärben ???",1,"Nix|Kill|Farbe")
227:               If But=3
228:                 Gosub Colget  ! neue Farbe oder
                                  Muster holen
229:                 Fl%(Eck_max+2,Rnr)=Cnr
                                         !abspeichern
230:                 Fl%(Eck_max+3,Rnr)=Czei
231:               Endif
232:               If But=2 And F_anz>0 !Fläche
                                         löschen???
233:                 For I=Rnr To F_anz !jawoll,durch
                                         Verschieben
234:                   For J=0 To Eck_max+3
                                        ! hinauswerfen
235:                     Fl%(J,I)=Fl%(J,I+1)
236:                   Next J
237:                 Next I
238:                 Dec F_anz    ! Flächenzähler -= 1
239:               Endif
240:             Endif
241:             Goto Starty ! und neue Box zeichnen..
242:           Else
243:             If I=0       ! Nach oben scrollen?
244:               Sub Stnr,16
245:               If Stnr<0
246:                 Stnr=0
247:               Endif
248:               Goto Starty
249:             Else
250:               Add Stnr,16 ! ansonsten nach unten
251:               If Stnr+16>F_anz
252:                 Stnr=F_anz-18
253:               Endif
254:               Goto Starty
255:             Endif
256:           Endif
257:         Endif
258:       Endif
259:       If Aktnr<>-1          ! Falls Fläche gewählt
260:         Gosub Flacker(Stnr+Aktnr-1) ! Die Punkte
                                          blinken
261:         Gosub Flacker(Stnr+Aktnr-1) ! lassen
262:       Endif
263:       Exit If Mmk=2    ! Bei Rechtsklick beenden
264:     Loop
265:   Endif
266: Return
267: '
268: Procedure Flacker(F_nr) ! Punktblinkroutine
269:   Local I,J,X,Y
270:   Graphmode 3
271:   For J=0 To Fl%(Eck_max+1,F_nr) ! Alle Punkte
272:     I=Fl%(J,F_nr)                ! umrechnen
273:     X=Ymi-Arbxyz%(0)+(Xyz%(0,I)+(Xyz%(2,I)
         -Arbxyz%(2))/Darstx)
274:     Y=Ymi-Arbxyz%(1)-(Xyz%(1,I)+(Xyz%(2,I)
         -Arbxyz%(2))/Darsty)
275:     If X>53 And X<Ys-3          ! und falls sichtbar
276:       Circle X,Y,4              ! anzeigen
277:     Endif
278:   Next J
279:   Graphmode 0
280: Return
281: '
282: Procedure Quader ! Quader in einem Zug erstellen
283:   Local P1,P2,P3,P4,P5,P6,P7,P8,I,Bo,A$
284:   A$="Quader erstellen. Wodurch|festlegen?
       (Mitte oder|Ecke unten links vorne)"
285:   Gosub Ali(2,A$,1,"Nix|Mitte|Ecke")
286:   If But>1
287:     Gosub Inbox("LÄNGE DER X,Y,Z-SEITEN|EINGEBEN.
            ",3)
288:     Bo=1
289:     For I=0 To 2
```

Listing geht weiter...

```
290:        Y%(I)=X%(I)              ! Werte merken...
291:        If Y%(I)=0
292:          Bo=0                   ! Falls 0 abbrechen
293:        Endif
294:      Next I
295:      If Bo
296:        Gosub Inbox("UND JETZT DEN BE-|FESTIGUNGS
                  PUNKT.",3)
297:        If But=2                 ! Falls Mittelpunkt
298:          For I=0 To 2
299:            X%(I)=X%(I)-Y%(I)/2 ! Un.li.vor. Ecke
                                      errechnen
300:          Next I
301:        Endif
302:        Gosub P_neu(X%(0),X%(1),X%(2)) ! Alle neuen
                                      Punkte anfordern
303:        P1=Pnr
304:        Gosub P_neu(X%(0),X%(1)+Y%(1),X%(2))
305:        P2=Pnr
306:        Gosub P_neu(X%(0)+Y%(0),X%(1)+Y%(1),X%(2))
307:        P3=Pnr
308:        Gosub P_neu(X%(0)+Y%(0),X%(1),X%(2))
309:        P4=Pnr
310:        Gosub P_neu(X%(0),X%(1),X%(2)+Y%(2))
311:        P5=Pnr
312:        Gosub P_neu(X%(0),X%(1)+Y%(1),X%(2)+Y%(2))
313:        P6=Pnr
314:        Gosub P_neu(X%(0)+Y%(0),X%(1)+Y%(1),
                  X%(2)+Y%(2))
315:        P7=Pnr
316:        Gosub P_neu(X%(0)+Y%(0),X%(1),X%(2)+Y%(2))
317:        P8=Pnr
318:        Gosub Fl4_neu(P1,P2,P3,P4) ! und die 6
                                      Flächen erstel-
                                      len
319:        Gosub Fl4_neu(P1,P5,P6,P2)
320:        Gosub Fl4_neu(P2,P6,P7,P3)
321:        Gosub Fl4_neu(P3,P7,P8,P4)
322:        Gosub Fl4_neu(P4,P8,P5,P1)
323:        Gosub Fl4_neu(P8,P7,P6,P5)
324:      Endif
325:    Endif
326:  Return
327:  '
328:  Procedure Fl4_neu(A,B,C,D)  ! 4-Punktfläche
                                      erstellen
329:    Fl%(0,F_anz)=A
330:    Fl%(1,F_anz)=B
331:    Fl%(2,F_anz)=C
332:    Fl%(3,F_anz)=D
333:    Fl%(Eck_max+1,F_anz)=3    ! alles eintragen
334:    Gosub Colget          ! Farbe oder Muster holen
335:    Fl%(Eck_max+2,F_anz)=Cnr ! abspeichern
336:    Fl%(Eck_max+3,F_anz)=Czei
337:    Inc F_anz             ! Flächenzähler erhöhen
338:  Return
```

Listing: 3D-Grafik-Edier- und Demoprogramm Teil 2

ENDE

Wann rufen Sie uns an?

Sie brauchen nur unsere Telefonnummer, denn wir denken, daß wir alles im Verkaufsprogramm haben, was es an Software für den ATARI ST gibt. Ehrlich!

Sehr verbraucherfreundlich: Wir geben auf alles Garantie!!!

Wir führen: Alle Software (Auch die Neueste), alle Bücher, alle Anwendungssoftware, fast ohne Ausnahme.

Neu in der Top Ten, schon bei uns erhältlich.

Hier ein Auszug aus unserem Programm:

Titel			
3 D-Galaxy		(dt)	59,90
19, Boot Camp		(dt)	64,95
1943	NEU		59,95
221 b Bakerstreet			79,90
500 cc Grand Prix			69,90
20.000 Meilen unter d. Meer		*(dt)	64,95
Action Service	NEU	(dt)	64,95
Alien Syndrome			59,95
Altair			59,90
Alternate Reality City		(dt)	69,90
Amazon			59,90
American Pool		(dt)	34,95
Arcade Force Four		(dt)	79,90
Arena			119,00
Arkanoid		(dt)	49,90
Arkanoid 2		(dt)	59,95
Artura	NEU	(dt)	64,95
Asterix im Morgenland		(dt)	64,95
Atax	NEU	(dt)	49,95
Autoduel			59,90
Backlash		(dt)	59,90
Bad Cat		(dt)	59,90
Balance of Power		*	99,00
Bard's Tale 1			89,00
Battleships		(dt)	49,95
Bermuda Project		(dt)	79,95
Better dead than Alien		(dt)	59,95
Beyond the Icepalace		(dt)	59,95
Beyond Zork		*	89,00
Bionic Commando		(dt)	59,95
Black Lamp			69,90
Blueberry: Das Gespenst...		(dt)	64,95
BMX-Simulator			49,90
Bob Morane Sience Fiction		(dt)	69,90
Bolo		*(dt)	69,00
Bolo Werkstatt	NEU	*(dt)	69,90
Bomb Jack			64,95
Bubble Bobble			59,90
Bubble Ghost		(dt)	59,95
Buggy Boy		(dt)	59,95
Captain America		(dt)	59,90
9Captain Blood		(dt)	74,95
Carrier Command		(dt)	84,95
Camonix Challenge		(dt)	69,90
Championship Baseball			89,90
Chese		*(dt)	89,90
Checkmate		*	39,90
Chopper X			29,95
Chronoquest	NEU	(dt)	99,90
Chubby Cristle		(dt)	64,95
Coruption		*(dt)	79,95
Clever und Smart		(dt)	59,95
Crazy Cars		(dt)	69,90
Cybernoids		(dt)	64,95
D.T. Olympic Challenge	NEU	(dt)	64,95
Dark Castle			89,00
Deathstrike		*(dt)	59,90
Defender of the Crown			79,90
Deja Vu			79,90
Dizzy Wizzard (1 MB RAM nötig)		*(dt)	59,90
Down at the Trolls	NEU	(dt)	59,95
Dr. Livingstone I Presume			49,90
Dungeon Master		(dt)	84,95
E C O (sphere)			59,90
Eden Blues		(dt)	89,00
Eddie Edwards Super Ski		(dt)	59,95
Electronic Pool		(dt)	69,90
Elemental	NEU	(dt)	59,95
Elite	NEU	(dt)	84,95
Empire Strikes Back			59,95
Enduro Racer			69,90
Epix Compilation			99,00
Euro Soccer 88		(dt)	59,95
Eye		(dt)	49,95
Fahrenheit 451		*	59,90
Fire Blaster		(dt)	39,90
Fire and Forget		(dt)	79,95
Football Fortunes			79,90
Fugger		(dt)	59,95
Fußball Manager 2		(dt)	64,95
Foundations Waste			79,95
Flight Simulator 2		*	179,00
Flight Sim. Scenery Japan			49,95
Flight Sim. Scenery Europa			49,95
Flight Sim. Scenery Disk 7			49,90
Flight Sim. Scenery Disk 11			49,90
Fred Feuerstein		(dt)	59,95
Freedom	NEU	(dt)	64,95
Gamblet		(dt)	49,90
Gato			99,00
Gary Lineker's Hotshots		(dt)	64,95
Gary Lineker's Superskills		(dt)	64,95
Gauntlet 2		(dt)	79,90
Gnome Ranger		*	49,90
Golden Path			69,90
Goldrunner 2		(dt)	59,90
Goldrunner 2-Scen. Disk 1			19,90
Goldrunner 2-Scen. Disk 2			19,90
Guild of Thieves		*	89,00
Hacker		*	69,95
Hardball			79,90
Harrier Strike Mission			89,00
Hellowoon		(dt)	69,95
Hollywood Hijinx		*	89,90
Hostages	NEU	(dt)	79,95
Hot Shot		(dt)	64,95
Impact			49,90
Impossible Mission 2			54,90
Intelligent Checkers		*(dt)	59,90
Indian Mission		*(dt)	59,90
Indoor Sports		(dt)	79,95
Inside Outing	NEU	(dt)	69,95
Jagd auf Roter Oktober		(dt)	84,95
Jet		*(dt)	119,00
Jinxter		*	79,95
Kampf um die Krone		(dt)	69,90
Kaiser		(dt)	129,00
Karting Grand Prix		(dt)	34,95
Killdozers		(dt)	59,95
Kings Quest 3er-Pack		*	84,95
Knight Orc		*	69,90
Leaderboard Birdie	NEU	(dt)	79,95
Leben und Sterben lassen	NEU	(dt)	64,95
Leisure Suit Larry			64,95
Legends of the Sword		*(dt)	79,95
Leviathan			49,90
Lombard RAC Ralley	NEU	(dt)	84,95
Luxor	NEU	(dt)	49,95
Lucky Luke: Nitroglyzerin		**(dt)	59,90
Mach 3			59,95
Marble Madness			89,90
Metropolis 2000 AD		(dt)	39,90
Mewilo		*(dt)	69,95
M G T		(dt)	89,00
Mickey Mouse		(dt)	59,95
Microleague Wrestling			69,90
Mindshadow		*	79,90
Moebius			79,90
Nebulus	NEU	(dt)	64,95
Netherworld		(dt)	64,95
Night Raider		(dt)	64,95
Nigel Mansell	NEU	(dt)	79,90
Nine Princess in Amber		*	59,90
Obliterator		(dt)	79,95
Off Shore Warrior	NEU	(dt)	59,95
Ogre			89,90
Oids		(dt)	59,90
Ooze		(dt)	79,95
Outcast		(dt)	39,90
Outrun		(dt)	59,90
Overlander	NEU	(dt)	64,95
Overlord	NEU		59,95
Pandora		(dt)	59,95
Pawn			89,90
Peter Pan	NEU	(dt)	64,95
Phantasm			59,95
Phantasie 3			79,90
Perry Mason		*	59,90
Pinball Factory			79,90
Pink Panther		(dt)	59,90
Plutos			59,90
Police Quest		*	64,95
Pool			34,95
Power Struggle		*	49,90
Q-Ball		(dt)	69,90
Quadralian		(dt)	59,90
Rana Rama			69,90
Return to Genesis		(dt)	59,90
Rings of Zilfin			79,90
Roadwars			69,90
Rockford			69,90
Rolling Thunder		(dt)	69,90
Rückkehr der Jedi Ritter	NEU	(dt)	64,95
Sargon Chess	NEU	(dt)	79,90
Seconds Out			69,90
Sentinel			69,95
Shackled		(dt)	59,95
Shuffleboard		(dt)	34,95
Shuttle 2			79,90
Side Walk		(dt)	69,90
Skyblaster		*(dt)	59,90
Sky Chess	NEU	(dt)	64,95
Skyfighter			49,90
Skyrider			69,90
Slapfight		(dt)	59,95
Slaygon Adventure		(dt)	69,90
Soldier of Light	NEU	(dt)	64,95
Sommer Olympiade		(dt)	64,95
Solomon's Key			59,90
Space Ace			69,90
Space Baller		(dt)	29,90
Space Harrier		(dt)	64,95
Space Port		(dt)	69,90
Space Quest 1		*	84,95
Space Quest 2		*	69,90
Space Station			79,90
Spitfire 40		(dt)	74,95
Spy versus Spy 1		(dt)	79,90
Starglider		*	89,90
Starglider 2		(dt)	84,95
Starray	NEU	(dt)	59,95
Startrash	NEU	(dt)	59,95



Titel			
Startreck		(dt)	64,65
Starwars			59,90
Stellar Crusade			99,90
Streetfighter		(dt)	59,90
Strike Force Harrier			89,00
Strip Poker 2		**(dt)	49,90
ST Soccer		(dt)	69,90
ST Wars		(dt)	79,90
Sub Battle			79,90
Sundog	NEU		49,90
Super Hang On	NEU	(dt)	64,95
Supersprint			49,90
Superstar Icehockey		(dt)	89,00
Taipan		(dt)	59,90
Tanglewood		(dt)	59,90
Tau Ceti		(dt)	59,95
Tetraquest	NEU		59,95
Tee Up Golf			59,90
Terramex		(dt)	59,95
Terrorpods			79,90
Test Drive		(dt)	89,00
Tetris			69,90
Thai Boxing		(dt)	59,90
Thrust			34,95
Thundercats		(dt)	59,95
Tracker		*	89,90
Trailblazer		(dt)	79,90
Trantor		(dt)	59,95
Trashheap		(dt)	69,90
Trivial Persuit	NEU	(dt)	64,95
Turbo			29,90
Ultima 3			89,00
Ultima 4			89,00
Uninvited			89,00
Universal Military Simulator		*	84,95
Vampire's Empire		(dt)	59,95
Vegas Gambler		(dt)	79,00
Vermeer		*/**(dt)	79,90
Veteran	NEU	(dt)	49,95
Virus		(dt)	64,95
Wanted	NEU	(dt)	64,95
War Games Construction Kit			79,90
War Hawk			34,95
Warship	NEU		99,95
Warzone			29,90
Waterskiing		(dt)	59,95
Western Games		(dt)	59,95
Where Times stood still	NEU	(dt)	64,95
Whriligig	NEU	(dt)	64,95
Winter Games		(dt)	79,90
Winter Olympiad '88		(dt)	64,95
Wizball		(dt)	59,90
Xenon			69,90
3 D Video Studio		(dt)	148,00
Advanced Art Studio			89,90
Alternative Keyboard Expander			79,90
Analysis		(dt)	79,90
Artscribe			89,90
BCPL			329,00
Cambridge Lisp			499,00
C Breeze	NEU		79,90
db Calc		(dt)	179,00
Flash Back			149,00
Font Expert			79,90
GST Editor			49,90
GST C-Compiler			249,00
Introduction to Logo			79,90
K-Communication V 2.0			179,00
K-Graph 2		(dt)	149,00
K-Ram			119,00
K-Resource		(dt)	129,00
K-Switch		(dt)	99,00
K-Spread			199,00
K-Spread 2		(dt)	229,00
Make			179,00
MCC-Pascal			299,00
Mighty Mail			79,90
Pascal Development V 2.0			299,00
Personal Money Manager			119,00
Quantum Paint		**	69,90
STOS Basic	NEU		89,90
T R I M		(dt)	349,00

Telefon: 0 24 72 - 71 58

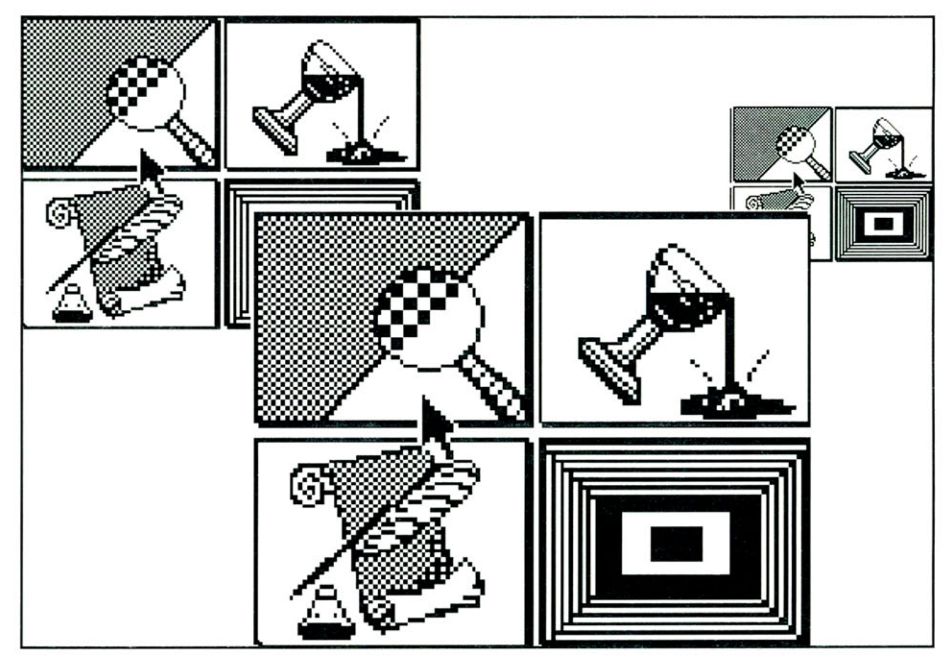
Odin-Software GmbH, Hengstbrüchelchen 39

D-5108 Monschau 4, Fax: 0 24 72 - 53 71

GROSSARTIG

Eine Echtzeitlupe im Selbstbau

Nachdem wir, ST-spezifisch gesehen, durch unsere Bildungsserie in den letzten Monaten eine kleine Pause eingelegt haben, kommen wir in dieser Ausgabe wieder zu den Wurzeln des STs zurück. Nach langer Tips & Tricks-Abstinenz haben wir uns etwas ganz Trickreiches ausgedacht und veröffentlichen hier eine LUPE, die in Echtzeit einen Ausschnitt des STs auf den vollen Bildschirm vergrößert. Dabei gehen wir nicht nur auf den Algorithmus der Lupe ein, sondern wagen uns auch an diverse Interrupts wie Maus, VBL und das XBIOS. Lassen Sie sich überraschen....



Zu allererst

Bitte haben Sie keine Furcht vor dem langen Listing, das sie vielleicht abtippen müßten. Da ich selbst ein Gegner davon bin (Wer tippt beispielsweise seitenlange DATA-Zeilen ab?), überlegte ich lange, ob wir dieses Listing veröffentlichen sollten. Wir taten es nun unter dem Aspekt, daß man, durch die Vielfalt der angesprochenen Problemlösungen, sehr gut daraus lernen kann, weil es genau dokumentiert ist und in dieser ST-Ecke erklärt wird. Ich möchte auch Assembleranfänger dazu aufrufen, sich dieses Programm unter den Nagel zu reißen, denn man findet ein paar sehr interessante Tricks und lernt dabei die etwas unbekannteren Befehle wie *TAS* oder *MOVEP* kennen. Sollten Sie das Programm tatsächlich auch verwenden, finden Sie wie immer den Quellcode sowie das ausführbare Pr ogramm auf der Monatsdiskette. Der Assemblercode wurde übrigens für den Metacomco-Assembler geschrieben, ist aber leicht auf den Digital-Research-Assembler umsetzbar. Trotzdem werden wir in Zukunft auch wieder zu kleineren Listings zurückkommen.

Aller guten Dinge sind drei

Sie werden im Laufe der ST-Ecke mitbekommen, daß einige Zeiger des Betriebssystems (XBIOS, Maus und VBL) ganz legal verbogen werden, um die Lupe sauber mit möglichst vielen Programmen laufen zu lassen. Genau diese eigenen Routinen sind neben den eigentlichen Lupenroutinen sehr interessant und sollen ausführlich erklärt werden - aber gehen wir vom Start des Programms aus:

Nicht so habgierig

Wird ein Programm gestartet, reserviert das GEMDOS prinzipiell erst einmal jeglichen Speicher (man kann ja nie genug kriegen), der dann bis auf den tatsächlich vom Programm benötigten wieder freigegeben werden muß. Da wir natürlich nicht den gesamten Speicher benötigen, berechnen wir die Länge unseres Programmes (inklusive Datenbereich) und übergeben das der GEMDOS-Routine *M_SHRINK*, die den reservierten Speicherblock auf die gewünschte Länge reduziert. Gesagt, getan, beginnen wir mit den weiteren Vorarbeiten: Cursor ausschalten, Copyright über unsere kleine einfache Routine *CONOUT* ausgeben.

A Kind of Magic

Als nächstes verbiegen wir die Maus-Interruptserviceroutine, deren Adresse sich in einer Struktur befindet, die wir über XBIOS' *KBDVBASE()* erfahren können (siehe ST-Ecke 'Der ST läßt das Mausen nicht...'). An der 16. Stelle (Byte) steht der Vektor der Mausroutine. Direkt vor unserer eigenen Maus-Interruptserviceroutine steht eine 'magic number'. Eine solche magische Nummer dient dazu, ganz spezielle Dinge (hier unsere eigene Mausroutine) eindeutig zu identifizieren. Bemerken wir, daß vor der aktuellen Mausroutine diese Nummer steht, so wissen wir, daß dies unsere Routine ist, die Lupe also schon einmal vorher gestartet wurde, und wir verabschieden uns sang- und klanglos. (Eine ähnliche und etwas ausführlichere Methode befindet sich am Beginn der VBL-Routine, aber dies erklären wir im Zusammenhang mit jener Routine.) War es der erste Start der Lupe, speichern wir den Einsprung der alten Mausroutine und fügen unsere Adresse ein.

Der kleine Unterschied

Ohne allzuviel vorwegzunehmen und zu verraten (der Trick mit der Lupe wird erst unten enthüllt), möchte ich hier den Unterschied zwischen der physikalischen und logischen Bildadresse kurz erläutern. Bei der physikalischen Adresse handelt es sich um die Anfangsadresse des Speicher-

bereichs, den Sie tatsächlich auch auf Ihrem Monitor zu sehen bekommen. Interessanterweise muß das aber nicht unbedingt die Adresse sein, die auch die Routinen benutzen, wenn zum Beispiel eine Linie gezogen oder ein Text geschrieben wird, was im allgemeinen über GEM und damit über *LINE-A* geschieht. Diese Routinen greifen nämlich auf die logische Bildadresse zu, so daß sie zum Beispiel in einen Bildbereich vom Programm aus zeichnen können, ohne daß es der Benutzer bemerkt. Nachdem dann Ihr Bild fertig ist, ändern Sie Ihre logische zur physikalischen Adresse. Klar, nicht? Da für unsere Lupe das Bild interessant ist, welches ursprünglich (also bevor es unsere nette Lupe gab) auf dem Bildschirm zu sehen war, holen wir uns die physikalische Adresse über *PHYSBASE* und speichern diese zwischen. Mit den nächsten Zeilen setzen wir die Adressen für unsere Bildspeicher. Die Lupe benötigt zwei Bildschirmspeicher: warum und wieso, das erfahren Sie unten, und deshalb wollen wir auch zunächst die Unterroutine *create_table* außen vor lassen (obwohl ich glaube, daß den ersten Lesern schon langsam ein Licht aufgeht, wie die Lupe vielleicht funktionieren könnte).

Startposition

Im Laufe des Programms greifen wir auf die Mauskoordinaten zu, um zu wissen, welcher Teil des Bildschirms um die Maus vergrößert werden soll. Dazu bekommen wir (siehe unten) relative Koordinaten vom Tastaturprozessor geliefert. Wir holen uns am Anfang über die AES-Routine *graf_mkstate* die momentanen absoluten Mauskoordinaten als Anfangskoordinaten und addieren im Mausinterrupt die relativen Koordinaten, da wir die AES-Routine im Interrupt nicht aufrufen können. Um auf *graf_mkstate* zugreifen zu können, haben wir uns natürlich als Applikation anzumelden.

Eine super Sache

Damit alles so richtig sauber funktioniert, müssen wir uns in den Vertical-Blank-Interrupt und in den XBIOS-TRAP einhängen. Ganz kurz für alle Einsteiger: Während der Monitor den Bildschirm darstellt, rast der Elektronenstrahl im Zickzack immer von rechts nach links und dabei langsam nach unten (siehe auch Bildwerkstatt ATARI ST 12/88 von Jörg Drücker und Alexander Beller). Ist er unten angekommen, wird er ausgeschaltet (Dunkeltastung) und rast wieder von der rechten unteren in die linke obere Ecke. In dieser Zeit, die man 'vertical blank' nennt, hat der Rechner Zeit, einige Dinge zu tun und bietet die Möglichkeit, ein paar kleine Routinen, die sich fein säuberlich anmelden, immer zu diesem Zeitpunkt auszuführen. Voraussetzung ist, daß die Gesamtheit aller VBL-Routinen die Zeit des VBL nicht überschreitet. Unsere Lupe ist eines dieser Programme und klinkt sich daher in die Kette der Programme ein. (Wie Sie unten sehen, werden wir sogar dafür sorgen, daß wir nur alle vier VBL-Aufrufe tatsächlich etwas tun, um den Rechner nicht allzusehr zu belasten.) Um uns in die Liste der Routinen einzutragen, die aufgerufen werden sollen, müssen wir auf eine Adresse zugreifen, die im vom Prozessor geschützten Speicherbereich liegt, was nur im SUPERVISOR-Modus geschehen darf. Falls der Benutzer die physikalische Bildschirmadresse ändert, wird er dies über eine XBIOS-Routine machen. Damit auch wir dies mitbekommen, hängen wir uns zusätzlich in den XBIOS-Einsprung, womit wir eine weitere Adresse im geschützten Speicherbereich ändern müssen. Das Ausführen der Routine INSTAL im Supervisormodus erreichen wir durch die Routine *SUPEXEC* des XBIOS.

ADE

Nach der Initialisierung aller Vektoren legen wir eine kleine Pause ein und verabschieden uns dann... aber halt: Wie Sie sicherlich schon erkannt haben, handelt es sich bei unserer Lupe um ein Programm, das resident im Speicher verankert wird und daher am Ende der Initialisierung nicht wieder entfernt werden darf. Dazu teilen wir dem Betriebssystem mit, welcher Speicherbereich nach dem Verlassen des Programms im Speicher liegenbleiben soll. Nachdem wir dies mit *KEEP_PROCESS* über GEMDOS getan haben, verabschiedet sich unser Programm endgültig, denn es wird ja nur ab und an gebraucht, nämlich dann, wenn ein VBL-Interrupt auftritt, die Maus angesprochen wird oder der Benutzer die physikalische Bildschirmadresse ändert.

Der XBIOS-TRAP

Schauen wir uns zunächst den XBIOS-Trap an. Alle Betriebssystemfunktionen werden über eine softgesteuerte Unterbrechung, einen sogenannten TRAP ausgelöst, dem wiederum eine Nummer zugeordnet wird. So hat zum Beispiel GEMDOS die Nummer 1 (TRAP #1), BIOS die Nummer 13 und XBIOS die 14. Die Adressen der daraufhin ausgeführten Routinen befinden sich ab Adresse $80 (plus Nummer, mal 4). Die Parameter der Routinen werden so auf den Stack gelegt, daß der letzte Parameter als erstes auf den Stack kommt. Die Nummer der Routine (*SUPEXEC* hat beispielsweise die Nummer $26) steht als letzter Parameter als Wort auf dem Stack. Nachdem die Routine durchgeführt worden ist, werden das Statusregister und der Programmzähler, die zuvor gerettet worden waren, wiederhergestellt, die Parameter befinden sich aber immer noch auf dem Stack, so daß der Stackpointer korrigiert werden muß. Wichtig ist, daß wir prüfen, ob das Programm, das den XBIOS-TRAP ausgelöst hat, im Supervisor-Modus war oder nicht, denn abhängig davon liegen die Parameter auf dem Supervisor- oder Userstack. Nachdem wir dies in unserer Routine durch Testen eines Bits im Statusregister geklärt haben, schauen wir, ob die Routinen *PHYSBASE* oder *SETSCREEN* vom Benutzer aufgerufen worden sind. Ist dies der Fall - dies erkennen wir an der Funktionsnummer -, merken wir uns den Parameter, der die physikalische Bildschirmadresse enthält und speichern ihn in der Variablen Physbase ab. Da man mit *SETSCREEN* auch andere Werte setzen und das Setzen eines bestimmten Parameters durch -1 verhindern kann, prüfen wir diese Tatsache zuvor ab. Danach springen wir in die Original-XBIOS-TRAP-Routine, was wir uns in der Initialisierung durch das Retten der Originaladresse in *XBIOS_VEC* ermöglicht haben.

Mäuschen hab' acht...

Wichtig für unsere Lupe ist es zu wissen, an welcher Stelle sich die Maus befindet. Zu diesem Zweck hängen wir uns in den Mausvektor. Was passiert nun in der Mausroutine, und wie können wir sie für unsere Zwecke nutzen? Im ATARI ST befindet sich ein eigener kleiner Prozessor (ein 6803 der MOTOROLA-6800-Familie, seines Zeichens Single-Chip-Processor), der für die Unterstützung der Tastatur, der Maus und des Joysticks zuständig ist. Die Kommunikation mit dem Rest der Welt geschieht über Interrupts und Pakete. Dadurch, daß die Maus durch einen Interrupt zur Ausführung kommt, funktioniert sie praktisch immer, auch dann, wenn die Floppy oder Harddisk angesprochen wird. [Bewegen Sie einmal auf dem PC (IBM & Co) die Maus, während ein Diskettenzugriff läuft: Zu-

nächst passiert nichts, und nach dem Zugriff holt die Maus das Versäumte nach...] Wenn nun ein solcher Interrupt ausgelöst wird, sendet der Chip ein Datenpaket von 3 Bytes, die wie folgt definiert sind:

Header:	Die gesetzten Bits des Wertes $F8 (11111000) kennzeichnen, daß es sich um ein Mauspaket handelt. Die beiden unteren Bits zeigen zusätzlich an, welche der beiden Tasten gedrückt worden sind. Daher muß man, um das Mauspaket zu erkennen, mit den Werten $f8 bis $fb rechnen.
DeltaX:	relative X-Mausposition, die vorzeichenbehaftet ist und damit zwischen -128 und +127 liegt
DeltaY:	relative Y-Mausposition; auch dieser Wert liegt zwischen -128 und +127.

Die Adresse des Mauspakets steht beim Einsprung in die Mausserviceroutine im Register A0! In unserer Mausroutine fragen wir auf den eben angesprochenen Header ab und speichern die beiden relativen Koordinaten, welche wir gleich zu unseren internen absoluten addieren. Dabei ist aber zu beachten, daß wir unsere absoluten Koordinaten nicht überschreiten - deshalb wird auf die entsprechenden Grenzen abgefragt. Das wäre auch schon fast alles, was es zu unserer Mausroutine zu sagen gäbe. Nebenbei: Es wäre übrigens ein leichtes, in diese Routine eine Verdoppelung der Mausgeschwindigkeit (sogenannte QUICKMOUSE) zu implementieren - man müßte nur die relativen Mauswerte manipulieren. Aber denken Sie daran, daß dies geschehen muß, bevor Sie selbst mit den Koordinaten arbeiten, denn auch unsere eigene Lupe soll diese Verdoppelung mitbekommen. Dies ist übrigens auch der Grund dafür, daß eine vorhandene Quickmouse-Utility vor der Lupe geladen werden muß, denn wenn die Mauskoordinaten nach dem Aufruf der Lupe verdoppelt werden, bemerkt dies die Lupe nicht, und die ST-internen Absolutkoordinaten der Maus laufen doppelt so schnell wie die unserer Lupe.

Statussymbole

Sicherlich ist es nicht wünschenswert, die ganze Zeit mit der Lupe zu arbeiten. Deshalb wurde ein Umschalter für die unterschiedlichen Modi (normal, 2fach-, 4fach-...Vergrößerung) eingebaut. Das Umschalten geschieht über die Tastenkombination ALTERNATE-SHIFT-SHIFT, welche wir natürlich im Programm irgendwie abfragen müssen. Der Status dieser Tasten ist als Bits kodiert in einer bestimmten Adresse zu finden - und genau damit beginnt das Problem. In älteren TOS-Versionen (vor 1.2) wurde diese Adresse nicht bekanntgegeben, so daß sich viele Programmierer auf die Suche begaben und eine Adresse fanden, in der der Status zu finden ist: $E1B. Wie das Leben so spielt, wurde diese NICHT-dokumentierte Adresse von ATARI ab der TOS-Version 1.2 geändert. Aber dafür gibt es ab sofort eine saubere Methode, die Adresse herauszubekommen, was aber, es sei noch einmal erwähnt, erst ab TOS-Version 1.2 funktioniert. An der Adresse $4f2 findet man die Adresse der Systemvariablen *_sysbase*, die auf den sogenannten Betriebssystemheader zeigt, in dem allerlei interessante Informationen zu finden sind. Da uns zwei Einträge interessieren, möchte ich ihn in Listing 1 vollständig als Struktur aufführen.

```
typedef struct
{
   int os_entry;    /* Einsprungadresse ins Betriebssystem */
   int os_version;  /* Betriebssystemversion */
   long os_start;   /* Startadresse des OS-Codes */
   long os_base;    /* Anfangsadresse des Betriebssystems */
   long os_memboot;/* Beginn des freien Speichers */
   long os_shell;   /* Shell */
   long os_magic;   /* Zeiger auf GEM-magic-Variable ->$8765432 */
   long os_gendat;  /* Erstellungsdatum des TOS im BCD-Format */
   int os_palmode  /* unterstes Bit gesetzt bedeutet PAL, sonst NTSC */

   int os_gendatg; /* Erstellungsdatum im GEMDOS-Format */
 /* folgende Einträge sind erst ab Version 1.2 des TOS verfügbar ! */
    long *_root; /* Zeiger auf Speicherblockliste(->GEMDOS INTERN
                    Sonderheft oder ST-Computer 9/87 - Alex Esser */
   long kbshift; /* Zeiger auf Adresse, die Tastaturstatus enthält */
   PD **_run; /* Zeiger auf PD des aktuellen Prozesses */
}SYSHDR;
```

Listing 1: Struktur des Betriebssystemheaders

Aus dieser Struktur greifen wir uns zwei Einträge heraus: zum einen *os_version* und zum anderen *kbshift*. In *os_version* schauen wir nach, ob es sich um ein Betriebssystem handelt, das die Versionsnummer 1.2 oder folgende hat. Ist dies der Fall, befindet sich im Header unter *kbshift* die Adresse des Tastaturstatus', die wir dann in unserem Programm benutzen. Ist die Versionsnummer kleiner, greifen wir auf die schon oben erwähnte Adresse $E1B zu. Wenn nun die Tasten ALTERNATE-SHIFT-SHIFT gedrückt werden, überprüfen wir das in jedem VBL-Interrupt und verwerten dies, indem wir einen Modus-Zähler erhöhen. Es ist allerdings wahrscheinlich, daß wir das Drücken und Wiederloslassen nicht in einer 70stel (oder 50stel) Sekunde schaffen, so daß der Zähler mehrfach erhöht werden würde, was natürlich nicht passieren darf! Wir merken uns den Status der Tasten des letzten VBLs und vergleichen ihn mit dem momentanen. Waren die Tasten im letzten VBL gedrückt, und der momentane Status ist jetzt gelöscht, sind die Tasten losgelassen worden, und wir erhöhen unseren Modus. Dieses Verfahren ist in seiner Art recht häufig anwendbar, deshalb sollten Sie es sich etwas genauer anschauen. Sie finden es in der Routine *GET_MODE*.

Unter der Lupe

Jetzt habe ich Sie aber lange genug neugierig gemacht! Da wir alle Vorarbeit geleistet haben, kommen wir endlich zur Hauptsache: der Lupe. Sie ist in verschiedenen Vergrößerungsstufen (und Auflösungen) realisiert worden. Da aber der Trick beziehungsweise das Schema der schnellen Vergrößerung praktisch dasselbe ist, wollen wir nur die Vierfach-Monochrom-Vergrößerung erklären.


STARKE
SOFTWARE

CPU-Schonung

Wenn die VBL-Routine angesprungen wird, prüft sie zunächst in der Systemvariablen *_flock*, die an der Adresse $43e liegt, nach, ob momentan ein Zugriff auf die Harddisk oder die Floppy (auf gut deutsch: DMA-Betrieb) vorliegt. Ist dies der Fall, verabschiedet sie sich, um einen solchen Peripheriezugriff nicht zu behindern. Andernfalls zählt sie einen Zähler bis zum Wert REPEAT hoch, der hier auf 4 gesetzt ist. Hat der Zähler den Wert erreicht, wird die Lupe aktiviert, so daß sie nur alle vier VBL-Einsprünge aktiv wird. Wie Sie beim Benutzen der Lupe feststellen werden, reicht das völlig aus, und wir belasten die CPU nicht allzu stark. Außerdem überprüfen wir unsere Auflösung, um die richtige Lupe einzusetzen. In *GET_RES* wird zusätzlich noch überprüft, ob unser(!) Vektor der Mausroutine noch gesetzt ist, oder ob das Betriebssystem nicht vielleicht seinen eigenen Vektor darübergeschrieben hat. Zeigt der Vektor wieder ins ROM, schreiben wir einfach unsere Adresse wieder hinein. Und nun geht's in die Lupe...

Prinzipiell beruht die Lupe darauf, daß sie den darzustellenden Bildschirm (also den an der physikalischen Adresse) in einer modifizierten Form an einen anderen Speicherplatz kopiert und dann den Videochip (liebevoll SHIFTER genannt) dazu veranlaßt, nicht den Speicher der physikalischen Adresse, sondern nun unseren Speicher darzustellen. Wie das funktionieren soll, wo doch die physikalische Adresse die Adresse ist, die dargestellt wird? Es ist ganz einfach: Das Betriebssystem benutzt diese 'physikalische Adresse' nur als Zwischenspeicher, wirklich maßgebend ist die Adresse, die im VIDEO-SHIFTER (dem Grafik-Chip des ST) steht. Vor jeder VBL schaut das Betriebssystem nach, ob sich der Wert in der physikalischen Adresse geändert hat. In diesem Fall schreibt es die neuen Werte in den Chip. Nur, was das Betriebssystem kann, können wir schon lange, oder? Wir schreiben also, nachdem wir unser modifiziertes Bild aufgebaut haben, in den Grafik-Chip des ST unsere Adresse. Das Schöne dabei ist, daß das Betriebssystem davon nichts bemerkt, denn seine physikalische Adresse hat sich ja nicht geändert, und somit schreiben alle Routinen brav weiterhin an die Originaladresse.

Größenwahn

Nachdem wir nun eine Möglichkeit gefunden haben, einen anderen Speicherbereich auf dem Monitor sichtbar zu machen als den, in den das laufende Programm schreibt, müssen wir noch eine Möglichkeit finden, möglichst schnell unseren Speicher mit dem Inhalt des Originalbildes zu füllen. Allerdings wäre das ja relativ langweilig, denn wir hätten damit nichts gewonnen. Wir wollen das Bild ja modifiziert oder, genauer, ausschnittsvergrößert darstellen. Beachten Sie, daß es dabei wirklich darauf ankommt, schnell zu sein, denn die VBL-Routine darf nicht ewig dauern. Wie also bringt man mit möglichst wenig Aufwand eine rasend schnelle Vergrößerung hin? Wir wollen uns das Prinzip in der monochromen Darstellung anhand der Vierfach-Vergrößerung anschauen. In der monochromen Auflösung wird ein Bildpunkt in einem Bit verschlüsselt. Acht Bildpunkte haben in einem Byte und die 640 Punkte einer Zeile haben in 80 Bytes Platz. Führen wir eine vierfache Vergrößerung durch, so bedeutet das, daß ein Punkt vier Bildpunkte in der Breite und Höhe hat, so daß er durch 4 Bits in einem Byte repräsentiert wird und über 4 Zeilen geht. Das wiederum bedingt, daß wir nur ein Viertel des Bildschirms gleichzeitig darstellen können (was bei einer vierfachen Vergrößerung zugegebenermaßen nicht weiter verwunderlich ist). Stellen wir uns einmal vor, der Punkt an der Koordinaten 0,0 wäre gesetzt und der an der Koordinate 1,0 nicht, so hätte das entsprechende Byte im Originalbild folgende Bitbelegung:

10xxxxxx : 1=gesetzt,
0=gelöscht,
x=nicht angegeben.

Vergrößern wir auf das Vierfache, ist ein Punkt 4 Bits breit, so daß die beiden Punkte nun ein Byte ausfüllen, was jetzt folgendermaßen aussieht: 11110000. Wichtig ist, daß auch die Bytes in den drei Zeilen darunter genauso ausschauen, denn unser Punkt ist ja nicht nur viermal so breit, sondern auch viermal so hoch! Gut, das Prinzip der Vergrößerung ist praktisch verstanden: Wir müsssen ein Bit auf vier Bits ausdehnen und eine Zeile auf vier. Nun ist das einzelne Setzen von Bits bei einem Umfang von 640x400 Punkten viel zu langsam, und wir müssen uns eine andere Idee ausdenken. Jedes Byte wird bei der Vierfachvergrößerung auf vier Bytes gedehnt, wobei es 256 Möglichkeiten eines Bytes gibt. Zur Verdeutlichung möchten wir ein paar Beispiele der Dehnungen darstellen (siehe Tabelle 1).

vorher (1 Byte)	nachher (4 Bytes)			
00000000	00000000	00000000	00000000	00000000
00000001	00000000	00000000	00000000	00001111
00000010	00000000	00000000	00000000	11110000
00100101	00000000	11110000	00001111	00001111
10111010	11110000	11111111	11110000	11110000
11111111	11111111	11111111	11111111	11111111

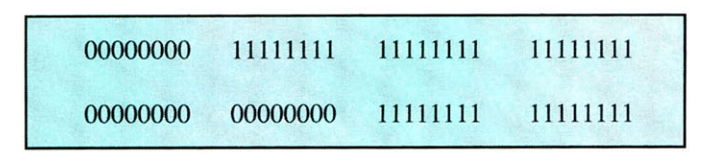
Tabelle 1: Beipiele der vierfachen Dehnung

Vergrößern wir unser Ausgangsbild, müßten wir für jedes Byte eine Dehnung durchführen und diese dann in das Zielbild kopieren. Aber wir können uns die Sache vereinfachen, indem wir die schon gedehnten Werte in einer Tabelle stehen haben. Stellen Sie sich die Tabelle wie die meiner Beispiele vor, nur mit dem Unterschied, daß sie vollständig ist und damit 256 Einträge à vier Bytes hat. Wenn ich beispielsweise das Byte mit dem Wert 50 habe, hole ich mir aus dem 50. Tabelleneintrag den schon gedehnten, vier Bytes langen Wert heraus, was praktisch keinen Rechenaufwand bedeutet, und kopiere ihn schnell in mein Zielbild. Das mache ich, ausgehend von der aktuellen X-Koordinate, mit zwanzig Bytes des Ausgangsbildes (Wir können nur 20*8 Bildpunkte darstellen, siehe oben.) und gehe zur nächsten Zeile des Quellbildes. Diese Zeile findet sich im Zielbild vierfach vergrößert vier Zeilen unter der letzten erzeugten Zeile. Haben wir alle Zeilen erzeugt, befinden sich noch Lücken zwischen den Zeilen des Zielbildes - wir haben ja nur eine Zeile erzeugt, also sind die drei dazwischen noch nicht gefüllt. Diese müssen wir nur noch von der Zeile obendrüber kopieren, wobei man durch einen besonderen Assemblertrick 40 Bytes auf einmal (!) kopieren kann!

Die Realisierung

Die Tabellen werden erst während des Programmablaufes erzeugt, was in der Routine *Create_table* geschieht. Den genauen Ablauf dieser vertrackten Routinen möchte ich hier nicht schildern, allerdings habe ich für diejenigen, die der Algorithmus interessiert, die Dokumen-

tation einer Tabellenerstellung besonders ausführlich gestaltet. Eine Verständnisfrage zu den Tabellen: Wie groß ist ein Eintrag einer Tabelle für den Vergrößerungsfaktor 8, und wie würde der Eintrag für den Bytewert $73 aussehen? (Erst überlegen:) Die Tabelle hat natürlich eine Eintragslänge pro Wert von acht Bytes und der Wert $73 (01110011) würde folgendermaßen aussehen:



Das Erstellen der 8er-Tabelle ist also sehr einfach, da ein Byte entweder nur ganz gefüllt oder gelöscht einem Bit entspricht.

Schauen wir uns nun als Beispiel *ZOOM_4* an, und bitte nehmen Sie sich für den nächsten Abschnitt etwas Zeit. Schauen Sie sich beim Lesen tatsächlich auch den Assemblercode an und versuchen Sie, ihn zu verstehen. Zunächst prüfen und setzen wir die Koordinaten des Lupenfensters. Danach holen wir die Adresse unserer Tabelle und setzen die Anzahl der zu kopierenden Zeilen auf 100 (400/4=100). Der Algorithmus besteht prinzipiell aus zwei Schleifen ineinander: Die äußere Schleife sorgt für hundert Zeilen (*dbra, d7, zoom_41*), die innere für die zwanzig Bytes einer Zeile (*dbra d6, zoom_42*), wobei immer zwei Bytes in einem Durchgang bearbeitet werden.

Das Erzeugen einer vergrößerten Zeile

Bei Label *ZOOM_42* holen wir uns zwei Bytes aus dem Originalbild. Bitte beachten Sie, daß die momentane X-Koordinate nicht unbedingt auf einer Bytegrenze liegen muß, deshalb werden die Bits auf die Wortgrenze geschoben. Durch das Schieben stehen die Bits nun im oberen Wort des Langworts, so daß wir diese wieder durch den *swap*-Befehl hinuntertransportieren müssen. Zunächst bearbeiten wir das obere Byte, indem wir es um 8 Bits nach unten verschieben. Danach benutzen wir es als Index in unsere Tabelle, wobei wir es wieder mit vier multiplizieren müssen. (Jeder Eintrag ist vier Bytes lang.) Da eine Multiplikation mit 4 einem zweifachen Schieben nach links gleichkommt, können wir das achtfache Schieben nach rechts und das 2fache nach links zu einer Operation '6faches Schieben nach rechts' zusammenfassen. Jetzt interessieren uns nur Werte von 0 bis 1024 (so lang ist unsere Tabelle) und zu diesem Zweck, blenden wir alle oberen Bits aus. Mit *move.l 0(a2,d3.w),(a1)+* addieren wir zu der Tabellenadresse in a2 den Index in *d3* (und 0), holen uns aus dieser Adresse unseren gedehnten Wert und schreiben ihn an die Zieladresse in *a1*. Das gleiche Spielchen führen wir nun mit dem zweiten im unteren Teil des Wortes befindlichen Byte durch, nur mit dem Unterschied, daß wir es vorher nicht um acht Stellen nach unten schieben, sondern um zwei Stellen nach oben (mal 4). Interessant ist, daß die Multiplikation hier durch die zweifache Addition mit sich selbst durchgeführt wird: Das ist schneller als ein zweifaches Schieben nach oben (links)! Haben wir die 20 Bytes hinter uns gebracht, ist die Zeile im Zielbild voll; bis zum Anfang des nächsten Bytes des Quellbildes fehlen aber noch 60 Bytes, die wir zur Quellbildadresse hinzuaddieren - im Zielbild überspringen wir die nächsten drei Zeilen(3*80), da wir diese am Ende durch ein reines Kopieren ergänzen. Sind nun alle 100 Zeilen vergrößert übertragen, wissen wir, daß im Zielbild immer noch Lücken zwischen den einzelnen Zeilen bestehen, die wir durch folgenden Assemblertrick sehr schnell zu füllen wissen:

Das Geheimnis des schnellen Kopierens

Es ist hinreichend bekannt, daß der MC68000-Prozessor, trotz seines wirklich variantenreichen Befehlssatzes, keinen Befehl zum Verschieben oder Kopieren von großen Blöcken besitzt. Es gibt aber einen Trick, mit dem man 40 Bytes auf einmal (ehrlich gesagt, in zwei Schritten) kopieren kann. Der Befehl '*movem.l*' ist der eigentlich dazu gedacht, eine Reihe von Registern auf einen Schlag abzuspeichern und wieder zu retten. Nun ist jedes Register 4 Bytes lang, und man kann es so einrichten, daß man 10 Register entbehrlich macht. Dadurch liest man mit '*movem.l (a0)+, d0-d5/a3-a6*' von der gewünschten Adresse *a0* in die Register *d0-d5* und *a3-a6* auf einmal 40 Bytes ein und erhöht dabei *a0* automatisch um vierzig. Abgespeichert werden die Bytes wieder mit dem komplementären Befehl '*movem.l d0-d5,a3-a6, Adresse*'. Um nun die drei freigebliebenen Zeilen zeilenweise nacheinander zu füllen, kopieren wir die Bytes mit diesem Trick sehr schnell, indem wir zur Zielzeile noch jeweils einen Offset von 40, 40+80 oder 40+160 addieren. Normalerweise würde man vermuten, daß man 80, 160 und 240 addierte, da aber der erste *movem.l das a0*-Register schon um vierzig erhöht hat, müssen wir vierzig vom Offset abziehen.

Zwei ist besser als Eins

Wenn Sie das Listing betrachten, werden Sie bemerken, daß das Programm mit zwei Bildschirmspeichern arbeitet und diese immer umschaltet. Das hat den Vorteil, daß man immer in den Speicher schreibt, den der Benutzer gerade nicht sieht. Daher bekommt er keine Kopierarbeiten zu sehen, und das Bild beginnt nicht zu flackern.

Das Ding mit der gespaltenen Adresse

Zum Schluß möchte ich Sie noch auf eine besondere Sache hinweisen. Oben wurde schon erwähnt, daß die Lupe im VBL die Bildschirmadresse im VIDEO-Shifter setzt. Die Adressen der BYTES dafür liegen an der Stelle $FF8201 und $FF8203. Jeder, der sich ein bißchen mehr auskennt, wird zwei Dinge bemerken: Erstens sind es nur zwei Bytes (für eine drei-Byte-Adresse), und zweitens liegen sie zu allem Unglück auch noch auf einer ungeraden Adresse. Das Problem der zwei Bytes erklärt sich, daß es beim ST nur möglich ist, den Bildschirm auf einer Adreßgrenze von 256 Bytes anfangen zu lassen. Das heißt, daß $FF8201 für das High-Byte und $FF8203 für das Mid-Byte der Adresse verantwortlich ist. Das Low-Byte wird vom Shifter automatisch intern auf Null gesetzt. Wären die Adressen nun auf der gerade Adresse zu finden, könnte man recht einfach die Werte hineinschreiben. Für den hier vorliegenden Fall bieten wir Ihnen eine in QUICK zu findende, elegante Lösung mit den Befehl '*movep*' an, der auf einmal die beiden Bytes eines Wortes mit einem Byte Abstand in den Speicher schreibt und das, bei einem Offset von Eins, sogar an eine ungerade Adresse.

XBRA

Wie schon erwähnt, haben wir eine weitere Möglichkeit in unser Programm eingebaut, an der wir erkennen können, 'daß wir wir sind'. Vor dem eigentlichen Einsprung der VBL-Routine steht eine Struk-

tur von drei mal vier Bytes. Im ersten Eintrag steht der Name *XBRA*, an dem man die Struktur erkennen kann. Im zweiten Eintrag findet sich der Name des Programms, das sich in die VBL eingeklinkt hat, wobei der Name aber nur vier Buchstaben lang sein darf. Im dritten Eintrag haben wir die Adresse, die ursprünglich als Einsprung in die VBL-Programmliste gegolten hat. Durch diese Struktur wäre es sogar einem anderen Programm möglich, unsere Routine wieder auszuhängen, indem sie unsere neue Adresse mit der in XBRA zu findenden Adresse überschreibt: Dadurch wären wir einfach aus der Kette entfernt worden.

Ich hoffe, daß wir Ihnen mit dieser Lupe einen kleinen Einblick in die vielfältigen Möglichkeiten des STs gegeben haben, und daß Sie von dem einen oder anderen Trick etwas lernen konnten. Lassen Sie sich nicht entmutigen, falls Sie Teile nicht verstanden haben, sondern lesen Sie sich den einen oder anderen Absatz ein weiteres Mal vor und schauen Sie Sie sich dazu die passende Stelle im Listing an. Und vergessen Sie nicht, die Lupe auch mal auszuprobieren: Es lohnt sich ganz bestimmt, versprochen....

(Stefan Höhn, Alexander Beller)

```
  1:
  2:  ************************************************
  3:  *
  4:  *
  5:  *           R E A L T I M E   M A G N I F I E R
  6:  *
  7:  *
  8:  *               Written by Alexander Beller
  9:  *
 10:  *             (C) Copyright by IMAGIC Grafik
 11:  *
 12:  *
 13:  ************************************************
 14:
 15:  GEMDOS:        equ  1     * Trapvektor GEMDOS
 16:  BIOS:          equ  13    * Trapvektor BIOS
 17:  XBIOS:         equ  14    * Trapvektor XBIOS
 18:
 19:  PRINT_LINE:    equ  $09   * GEMDOS Funktionen
 20:  KEEP_PROCESS:  equ  $31
 21:  M_SHRINK:      equ  $4A
 22:
 23:  GET_PHYSBASE:  equ  $02   * XBIOS Funktionen
 24:  GET_RESOLUTION: equ  $04
 25:  SET_SCREEN:    equ  $05
 26:  SET_CURSOR:    equ  $15
 27:  GET_KBDVBASE:  equ  $22
 28:  GO_SUPER:      equ  $26
 29:
 30:  REPEAT:        equ  4     * Wiederholfrequenz für
 31:  *                         * den Bildaufbau im VBL
 32:
 33:  ************************************************
 34:
 35:  PROGRAM_START:
 36:      move.l    4(sp),a6
 37:      lea       USER_STACK,sp      * Eigenen STACK
                                       installieren
 38:*
 39:      pea       (USER_STACK-BSS_START)
                    +(TEXT_END-PROGRAM_START)+256
 40:      pea       (a6)
 41:      clr.w     -(sp)
 42:      move.w    #M_SHRINK,-(sp)    *Nicht benötigten
                                       Speicher
 43:      trap      #GEMDOS            * ans System
                                       zurückgeben
 44:      lea       12(sp),sp
 45:*
 46:  ***************************************
 47:*
 48:      clr.l     -(sp)
 49:      move.w    #SET_CURSOR,-(sp)  * Cursor
                                       ausschalten
 50:      trap      #XBIOS
 51:      addq.l    #6,sp
 52:*
 53:      pea       MESSAGE(pc)        * Copyright
                                       ausgeben
 54:      bsr       CONOUT
 55:
 56:  ***************************************
 57:*
 58:      move.w    #GET_KBDVBASE,-(sp) * Adresse der
 59:      trap      #XBIOS             * Tastatur-Vektor-
                                       tabelle holen
 60:      addq.l    #2,sp
 61:      move.l    d0,a0
```

```
 62:*
 63:      lea       KBDVBASE(pc),a1
 64:      move.l    a0,(a1)+       * Adresse der
                                   * Vektortabelle sichern
 65:      move.l    16(a0),a2            * Aktueller Maus-
                                         Interruptvektor
 66:*
 67:      cmpi.l    #$02101968,-4(a2) * Magic-Number vor
                                      der Maus-Routine?
 68:      beq       EXIT                 * Dann ist Lupe
                                       schon installiert
 69:*
 70:      ****************************************
 71:*
 72:      move.l    a2,(a1)+            * Alten Mausvektor
                                          sichern
 73:      move.l    #NEW_MOUSE,16(a0) * Neuen Mausvektor
                                          setzen
 74:*
 75:      move.w    #GET_PHYSBASE,-(sp)* Physikalische
                                          Bildschirm-
 76:      trap      #XBIOS              * adresse holen
 77:      addq.l    #2,sp
 78:      move.l    d0,PHYSBASE
 79:*
 80:      move.l    #FLIPSCR_1+255,d0   * Basisadresse
                                          von Bild 1/2
 81:      clr.b     d0                  * auf einer 256-
                                          Byte-Grenze
 82:      move.l    d0,FLIPBASE_1       * Adresse von
                                          Bild1 setzen
 83:      add.l     d0,FLIPBASE_2       * Adresse von
                                          Bild2 setzen
 84:*
 85:      bsr       APP_INIT            * Application
 86:*                                   * anmelden
 87:      bsr       GRAF_MKSTATE        * Mausposition
 88:*                                   * holen
 89:      bsr       CREATE_TABLE        * Tabellen
 90:*                                   * erstellen
 91:      pea       INSTAL(pc)          *Supervisor-Unter
                                        programm aufrufen
 92:      move.w    #GO_SUPER,-(sp)     * (XBIOS- u.VBL-
                                        Vektor umleiten)
 93:      trap      #XBIOS
 94:      addq.l    #6,sp
 95:*
 96:      bsr       PAUSE   * Pause einlegen und danach
 97:*                        * Programm resident beenden
 98:      clr.w     -(sp)
 99:      pea       (BSS_END-BSS_START)
                    +(TEXT_END-PROGRAM_START)+256
100:      move.w    #KEEP_PROCESS,-(sp)
101:      trap      #GEMDOS
102:*
103:  ****************************************
104:*
105:  EXIT: bsr       PAUSE             * Pause einlegen
106:*
107:        clr.w     -(sp)             * Programm verlassen
108:        trap      #GEMDOS
109:*
```

Listing geht weiter

```
110:  ******************************************
111:  *
112:  *          XBIOS-Trap
113:  *
114:  ******************************************
115:*
116:  NEW_XBIOS:
117:      lea       6(sp),a0     * Zeiger auf SUPERVISOR-
                                   Parameter
118:      move.w    (a0),d0      * Funktionsnummer vom SSP
                                   holen
119:*
120:      btst      #5,(sp)      * Aufruf vom SUPERVISOR?
121:      bne.s     SVR_CALL     * Ja....
122:*
123:      move.l    usp,a0       * Nein, Zeiger auf USER-
                                   Parameter
124:      move.w    (a0),d0      * Funktionsnummer vom USP
                                   holen
125:*
126:  SVR_CALL:
          cmpi.w    #GET_PHYSBASE,d0     * Funktion
                                           GET_PHYSBASE?
127:      beq.s     F_PHYSBASE           * Ja...
128:
129:      cmpi.w    #SET_SCREEN,d0       * Funktion
                                           SET_SCREEN?
130:      beq.s     F_SETSCREEN          * Ja...
131:*
132:  SYS_XBIOS:
133:      move.l    XBIOS_VEC(pc),-(sp) * Zum SYSTEM-
                                          TRAP springen
134:      rts
135:*
136:  F_PHYSBASE:
137:      move.l    PHYSBASE(pc),d0 * Adresse des
                                      Hintergrundbildes
138:      rte                         * zurückliefern
139:*
140:  F_SETSCREEN:
141:      move.l    6(a0),d0          * Soll Physbase
                                        umgesetzt werden?
142:      bmi.s     SYS_XBIOS         * Nein...
143:*
144:      move.l    d0,PHYSBASE       * Neuen Zeiger
                                        sichern
145:      move.w    #-1,6(a0)         * Physbase NICHT
                                        umsetzen
146:      bra.s     SYS_XBIOS
147:*
148:  ******************************************
149:  *
150:  *          VBL Interrupt
151:  *
152:  ******************************************
153:*
154:  XBRA:
          dc.b      'xbra'       * gross oder KLEIN?
155:      dc.l      04031965
156:  XBRA_VBL:
          dc.l      NEW_VBL      * Adresse des eigenen VBL
157:*
158:  NEW_VBL:
          movem.l   d0-d2/a0-a2,-(sp) * Register retten
159:*
160:      move.l    $45E,d0       * Screenzeiger gesetzt?
161:      beq.s     NO_SCREENPTR  * Nein...
162:*
163:      clr.l     $45E          * Screenzeiger löschen
164:      move.l    d0,PHYSBASE   * und neuen Bildvektor
                                    übernehmen
165:*
166:  NO_SCREENPTR:
167:      tst.w     $43E          * Wenn Floppy oder
                                    Harddisk aktiv, dann
168:      bne.s     GO_SYSTEM     * nichts weiter tun
169:*
170:      bsr       GET_MODE      * Modus holen
171:*
172:      bsr.s     GET_RES       * Auflösung holen
173:*
174:      lea       COUNTER(pc),a0 * Zähleradresse
175:      subq.w    #1,(a0)       * Countdown abgelaufen?
176:      bpl.s     GO_SYSTEM     * Nein....

177:*
178:      move.w    #REPEAT,(a0) * Zähler neuladen
179:*
180:      bsr.s     LUPE          * Bild aufbauen
181:*
182:  GO_SYSTEM:
183:      movem.l   (sp)+,d0-d2/a0-a2 * Register laden
184:*
185:      move.l    VBL_VEC(pc),-(sp) * Weiter mit dem
                                        System-VBL
186:      rts
187:*
188:  ******************************************
189:*
190:  GET_RES:
          moveq     #3,d0             * Komplett neu
191:      and.b     $FFFF8260,d0      * Auflösung in d0
                                        und sichern
192:*
193:      cmp.w     RESOLUTION(pc),d0
194:      beq.s     CMP_MOUSE
195:*
196:      move.w    d0,RESOLUTION
197:      clr.b     MODE
198:*
199:  CMP_MOUSE:
200:      lea       NEW_MOUSE(pc),a1 * Adresse der
                                       eigenen Routine
201:*
202:      move.l    KBDVBASE(pc),a0 * Wurde Mausvektor
                                      umgesetzt?
203:      move.l    16(a0),d0     * Aktuellen Vektor holen
204:*
205:      cmp.l     a1,d0         * Mit eigenem Vektor
                                    vergleichen
206:      beq.s     NO_CHANGE     * Ist noch der alte...
207:*
208:      moveq     #23,d1
209:      btst      d1,d0
210:      beq.s     NO_CHANGE     * Wenn nicht, dann
                                    alles so lassen...
211:*
212:      move.l    d0,KBDVBASE+4 * Vektor sichern
213:      move.l    a1,16(a0)     * Eig. Vektor setzen
214:*
215:  NO_CHANGE:
216:      rts
217:*
218:  ******************************************
219:*
220:  LUPE:
          move.l    PHYSBASE(pc),a0   * Hintergrundbild
221:      move.l    FLIPBASE_1(pc),a1 * Aktuelles
                                        Zielbild
222:*
223:      move.b    MODE(pc),d0       * Modus holen
224:      beq       QUICK             * Keine
                                        Vergrösserung
225:*
226:      movem.l   d3-d7/a3-a6,-(sp) * Register retten
227:*
228:      pea       LUPE_END(pc)      * Return-Adresse
                                        auf den Stack
229:*     erleichtert das bedingte Rufen
230:*     von Unterprogrammen
231:*
232:      move.w    RESOLUTION(pc),d1 * Auflösung holen
233:*
234:      beq       ZOOM_LOW      * Niedere Auflösung
235:*
236:      subq.w    #1,d1         * Mittlere Auflösung
237:      beq       ZOOM_MEDIUM
238:*
239:      subq.b    #1,d0         * High-Res Faktor 8
240:      beq       ZOOM_8
241:*
242:      subq.b    #1,d0         * High-Res Faktor 4
243:      beq       ZOOM_4
244:*
245:      bra       ZOOM_2        * High-Res Faktor 2
246:*
247:*
248:  LUPE_END:
          move.l    FLIPBASE_1(pc),d1  * Bildadressen
                                         tauschen
249:      move.l    FLIPBASE_2(pc),d0
250:      movem.l   d0-d1,FLIPBASE_1
251:*
252:      movea.w   #$8200,a1          * Adresse setzen
```

Listing geht weiter

```
253:      lsr.l      #8,d1
254:      movep.w    d1,1(a1)
255:*
256:      movem.l    (sp)+,d3-d7/a3-a6  * Register laden
257:      rts
258:*
259:  ******************************************
260:*
261:  ZOOM_8:
          moveq      #40,d2        * Aus der Grösse des
                                     Lupenfensters
262:      moveq      #25,d3        * wird der Zeiger ins
                                     Hintergrund-
263:      move.w     #559,d4       * bild berechnet
264:      move.w     #349,d5
265:      bsr        CALC_SCRPTR
266:*
267:      adda.w     d0,a0         * Offsets auf die
                                     Adresse des
268:      adda.w     d1,a0         * Hintergrundbildes
                                     addieren
269:*
270:      lea        BUFFER_8,a2   * Zeiger auf Tabelle
271:      move.w     #255*8,d5     * Maske für den
                                     Tabellenindex
272:      moveq      #49,d7        * 50 Zeilen pro
273:*                              * Bildschirm
274:  ZOOM_81:
          moveq      #4,d6         * 5 Words pro Zeile
275:*
276:  ZOOM_82:
          move.l     (a0),d0       * Daten aus Hinter-
                                     grundbild holen
277:      addq.l     #2,a0         * Stets 2 Bytes
                                     pro Durchgang
278:*
279:      rol.l      d2,d0         * Bytes auf Wortgrenze
                                     shiften
280:      swap       d0            * Bytes nach "unten"
                                     holen
281:      move.w     d0,d1         * Ergebnis sichern
282:*
283:      lsr.w      #5,d0         * Byte 1 nach "rechts"
                                     schieben
284:*                              * und mit 8 multi-
                                     plizieren-> Index
285:*
286:      and.w      d5,d0             * Index maskieren
287:      move.l     0(a2,d0.w),(a1)+ * Langwort 1 aus
                                         Tabelle holen
288:      move.l     4(a2,d0.w),(a1)+ * Langwort 2 aus
                                         Tabelle holen
289:*
290:      lsl.w      #3,d1             * Byte 2 * 8
                                         -> Index
291:*
292:      and.w      d5,d1             * Index maskieren
293:      move.l     0(a2,d1.w),(a1)+ * Langwort 1 aus
                                         Tabelle holen
294:      move.l     4(a2,d1.w),(a1)+ * Langwort 2 aus
                                         Tabelle holen
295:*
296:      dbra       d6,ZOOM_82    * Nächst.Word bearbeiten
297:*
298:      lea        70(a0),a0     * Zeiger auf die jeweils
299:      lea        7*80(a1),a1   * nächste Zeilen setzen
300:*
301:      dbra       d7,ZOOM_81    * Nächst.Zeile
302:*                              * bearbeiten
303:  ******************************************
304:*
305:      move.l     FLIPBASE_1(pc),a0    * Zieladresse
306:      moveq      #49,d7    * 100 Abschnitte à 4 Zeilen
307:*
308:*
309:*
310:  FILL_10:
          moveq      #1,d6     * 2 x 40 Bytes pro Zeile
311:*
312:  FILL_11:
          movem.l    (a0)+,d0-d5/a3-a6    * 40 Bytes nach
                                            unten kopieren
313:      movem.l    d0-d5/a3-a6,40(a0)
314:      movem.l    d0-d5/a3-a6,40+80(a0)
315:      movem.l    d0-d5/a3-a6,40+160(a0)
316:      movem.l    d0-d5/a3-a6,40+240(a0)
317:      movem.l    d0-d5/a3-a6,40+320(a0)
318:      movem.l    d0-d5/a3-a6,40+400(a0)
319:      movem.l    d0-d5/a3-a6,40+480(a0)
320:      dbra       d6,FILL_11
321:*
322:      lea        7*80(a0),a0    * Zeiger auf nächste
                                      Zeile setzen
323:      dbra       d7,FILL_10     * und weiter...
324:*
325:      rts
326:*
327:  ******************************************
328:  ZOOM_4:
329:      moveq      #80,d2          * Aus der Grösse des
                                       Lupenfensters

330:      moveq      #50,d3          * wird der Zeiger ins
                                       Hintergrund-
331:      move.w     #479,d4         * bild berechnet
332:      move.w     #299,d5
333:      bsr        CALC_SCRPTR
334:*
335:      adda.w     d0,a0           * Offsets auf die
                                       Adresse des
336:      adda.w     d1,a0           * Hintergrundbildes
                                       addieren
337:*
338:      lea        BUFFER_4,a2     * Zeiger auf Tabelle
339:      moveq      #99,d7          * Anzahl Zeilen
340:      move.w     #4*255,d1       * Maske für Tab.index
341:      moveq      #6,d5           * Konstante
342:*
343:  ZOOM_41:
          moveq      #9,d6           * 10 Words pro Zeile
344:*
345:  ZOOM_42:
          move.l     (a0),d0         * Daten aus  Hinter-
                                       grundbild holen
346:      addq.l     #2,a0           * Stets 2 Bytes
                                       pro Schleife
347:*
348:      rol.l      d2,d0  * Bytes: Wortgrenze shiften
349:      swap       d0     * Bytes nach "unten" bringen
350:*
351:      move.w     d0,d3  * Byte 1 "rechts" schieben
352:      ror.w      d5,d3             * mal 4 -> Index
353:      and.w      d1,d3             * Index maskieren
354:      move.l     0(a2,d3.w),(a1)+  * Langwort aus
                                         Tabelle holen
355:*
356:      add.w      d0,d0
357:      add.w      d0,d0
358:      and.w      d1,d0             * 2. Byte holen
359:      move.l     0(a2,d0.w),(a1)+  * Langwort aus
                                         Tabelle holen
360:*
361:      dbra       d6,ZOOM_42
362:*
363:      lea        60(a0),a0
364:      lea        240(a1),a1
365:*
366:      dbra       d7,ZOOM_41
367:*
368:  ******************************************
369:*
370:      move.l     FLIPBASE_1(pc),a0 * Zieladresse
371:      moveq      #99,d7            * 100 Abschnitte
                                         zu 4 Zeilen
372:*
373:  FILL_2:
          movem.l    (a0)+,d0-d5/a3-a6 * 40 Bytes nach
                                         unten kopieren
374:      movem.l    d0-d5/a3-a6,40(a0)
375:      movem.l    d0-d5/a3-a6,40+80(a0)
376:      movem.l    d0-d5/a3-a6,40+160(a0)
377:*
378:      movem.l    (a0)+,d0-d5/a3-a6 * 40 Bytes nach
                                         unten kopieren
379:      movem.l    d0-d5/a3-a6,40(a0)
380:      movem.l    d0-d5/a3-a6,40+80(a0)
381:      movem.l    d0-d5/a3-a6,40+160(a0)
382:*
383:      lea        3*80(a0),a0       * Zeiger auf
                                         nächste Zeile
384:      dbra       d7,FILL_2         * und weiter...
385:*
386:      rts
```

Listing geht weiter

```
387:*
388:*******************************************
389:*
390:  ZOOM_2:
        move.w     #160,d2  * Hohe Auflösung Faktor 2
391:    moveq      #100,d3  * Alles neu ....
392:    move.w     #319,d4
393:    move.w     #199,d5
394:    bsr        CALC_SCRPTR
395:*
396:    adda.w     d0,a0    * Offsets auf Adresse vom
397:    adda.w     d1,a0    * Hintergrundbild addieren
398:*
399:    lea        80(a1),a2
400:    lea        BUFFER_2,a3
401:*
402:    move.w     #199,d7  * Anzahl Zeilen
403:    move.w     #$1FE,d1 * Maske
404:    moveq      #7,d5    * Konstante

405:*
406:  ZOOM_21:
        moveq      #19,d6   * 20 Words pro Zeile
407:    move.w     (a0)+,d0 * Erstes Word holen
408:*
409:  ZOOM_22:
        swap       d0
410:    move.w     (a0)+,d0  * Sourcedaten holen
411:    move.l     d0,d4     * Daten in d0 erhalten
412:*
413:    lsl.l      d2,d4     * Word zurechtshiften
414:    swap       d4        * Zweites Byte in Byte 4
415:*                        * Erstes Byte in Byte 3
416:*
417:    move.w     d4,d3     * Erstes Byte holen
418:    lsr.w      d5,d3
419:    and.w      d1,d3
420:    move.w     0(a3,d3.w),d3
421:    move.w     d3,(a1)+
422:    move.w     d3,(a2)+
423:*
424:    add.w      d4,d4
425:    and.w      d1,d4     * Zweites Byte holen
426:    move.w     0(a3,d4.w),d4
427:    move.w     d4,(a1)+
428:    move.w     d4,(a2)+
429:*
430:    dbra       d6,ZOOM_22
431:*
432:    lea        40-2(a0),a0
433:    lea        80(a1),a1
434:    lea        80(a2),a2
435:*
436:    dbra       d7,ZOOM_21
437:*
438:    rts
439:*
440:  *******************************************
441:  *
442:  ZOOM_MEDIUM:
443:    move.w     #160,d2        * Mittlere Auflösung
                                    Faktor 2
444:    moveq      #50,d3
445:    move.w     #639-2*160,d4
446:    moveq      #199-2*50,d5
447:    bsr        CALC_SCRPTR
448:*
449:    add.w      d1,d1          * Offsets an mittlere
450:    add.w      d0,d0          * Auflösung anpassen
451:    add.w      d0,d1        * Offsets auf Adresse vom
452:    adda.w     d1,a0       * Hintergrundbild addieren
453:*
454:    lea        160(a1),a2     * Zeiger auf
                                    Hintergrundzeile+1
455:    lea        BUFFER_2,a3    * Tabellenzeiger
456:*
457:    moveq      #99,d7         * Anzahl Zeilen
458:    move.w     #$1FE,d1       * Maske
459:    moveq      #7,d5          * Konstante
460:*
461:  ME2_10:
        moveq      #19,d6         * 20 Words pro Zeile
462:*
463:  ME2_20:
        moveq      #1,d0          * 2 Planes bearbeiten
464:*
465:  ME2_21:
        move.w     (a0)+,d4   * Sourcedaten holen
466:    swap       d4
467:    move.w     2(a0),d4
468:*
469:    lsl.l      d2,d4      * Word zurechtshiften
470:    swap       d4         * Zweites Byte in Byte 4
471:*                         * Erstes Byte in Byte 3
472:*
473:    move.w     d4,d3      * Erstes Byte holen
474:    lsr.w      d5,d3
475:    and.w      d1,d3
476:    move.w     0(a3,d3.w),d3
477:    move.w     d3,(a1)+
478:    move.w     d3,(a2)+
479:*
480:    add.w      d4,d4
481:    and.w      d1,d4      * Zweites Byte holen
482:    move.w     0(a3,d4.w),d4
483:    move.w     d4,2(a1)
484:    move.w     d4,2(a2)
485:*
486:    dbra       d0,ME2_21
487:*
488:    addq.l     #4,a1
489:    addq.l     #4,a2
490:*
491:    dbra       d6,ME2_20
492:*
493:    lea        80(a0),a0  * Zeiger auf die jeweils
                                nächsten
494:    lea        160(a1),a1 * Zeilen setzen
495:    lea        160(a2),a2
496:*
497:    dbra       d7,ME2_10
498:*
499:    rts
500:*
501:  *******************************************
502:*
503:  ZOOM_LOW:
        moveq      #80,d2    * Low-Res Faktor 2
504:    moveq      #50,d3
505:    move.w     #319-2*80,d4
506:    moveq      #199-2*50,d5
507:    bsr        CALC_SCRPTR
508:*
509:    add.w      d1,d1  * Offsets an niedere
510:    lsl.w      #2,d0  * Auflösung anpassen
511:    add.w      d0,d1  * Offsets auf Adresse des
512:    adda.w     d1,a0  * Hintergrundbildes addieren
513:*
514:    lea        160(a1),a2
515:    lea        BUFFER_2,a3
516:*
517:    moveq      #99,d7     * Anzahl Zeilen-1
518:    move.w     #$1FE,d1   * Maske
519:    moveq      #7,d5      * Konstante

520:*
521:  LO2_10:
        moveq      #9,d6      * 10 Words pro Zeile
522:*
523:  LO2_20:
        moveq      #3,d0      * 4 Planes bearbeiten
524:*
525:  LO2_21:
        move.w     (a0)+,d4   * Sourcedaten holen
526:    swap       d4
527:    move.w     6(a0),d4
528:*
529:    lsl.l      d2,d4      * Word zurechtshiften
530:    swap       d4         * Zweites Byte in Byte 4
531:*                         * Erstes Byte in Byte 3
532:*
533:    move.w     d4,d3      * Erstes Byte holen
534:    lsr.w      d5,d3
535:    and.w      d1,d3
536:    move.w     0(a3,d3.w),d3
537:    move.w     d3,(a1)+
538:    move.w     d3,(a2)+
539:*
540:    add.w      d4,d4
541:    and.w      d1,d4      * Zweites Byte holen
542:    move.w     0(a3,d4.w),d4
543:    move.w     d4,6(a1)
544:    move.w     d4,6(a2)
```

Listing geht weiter

```
545:
546:        dbra      d0,LO2_21      * Nächste Einzelplane
                                       bearbeiten
547:*
548:        addq.l    #8,a1
549:        addq.l    #8,a2
550:
551:        dbra      d6,LO2_20      * Nächste 4 Planes
                                       bearbeiten
552:*
553:        lea       80(a0),a0      * Zeiger auf nächste
554:        lea       160(a1),a1     * Zeilen setzen
555:        lea       160(a2),a2
556:*
557:        dbra      d7,LO2_10
558:*
559:        rts
560:*
561:   ******************************************
562:*
563:   QUICK:
        movea.w   #$8200,a1   * Hardwareadresse
564:        move.l    a0,d0       * Hintergrundbild
565:*
566:        lsr.l     #8,d0
567:        movep.w   d0,1(a1)    * Bildadresse setzen
568:*
569:        rts
570:*
571:   ******************************************
572:   *
573:   CALC_SCRPTR:
574:        movem.w   MOUSE_POS(pc),d0-d1 * Mausposition
                                            holen
575:*
576:        sub.w     d2,d0          * Linke Lupenkante
577:        bpl.s     CHECK_1        * Ok....
578:        moveq     #0,d0
579:*
580:   CHECK_1:
        sub.w     d3,d1          * Obere Lupenkante
581:        bpl.s     CHECK_2        * Ok....
582:        moveq     #0,d1
583:*
584:   CHECK_2:
        cmp.w     d4,d0          * Rechte Lupenkante
585:        bls.s     CHECK_3        * Ok.....
586:        move.w    d4,d0
587:*
588:   CHECK_3:
        cmp.w     d5,d1       * Untere Pixelkante
589:        bls.s     CHECK_4     * Ok....
590:        move.w    d5,d1
591:*
592:   CHECK_4:
        moveq     #15,d2      * Offset in Punkten auf
                                die nächste links
593:        and.w     d0,d2       * liegende Wortgrenze
594:*
595:        lsr.w     #4,d0       * Offset in Bytes auf
                                Wortgrenze
596:        add.w     d0,d0
597:        mulu      #80,d1      * Offset in Bytes /Zeile
598:        rts
599:   ******************************************
600:*
601:*  In der nächsten Routine wird der Status
602:*  der Tasten abgefragt.
603:*  Entsprechend der Tasten und der Auflösung
604:*  wird daraufhin umgeschaltet.
605:*  Bei Farbe werden die Vektoren neu gesetzt, da
606:*  sie beim 'Reset' überschrieben werden!
607:*
608:   GET_MODE:
        move.l    SHIFT_ADDR(pc),a0 * Shiftstatus
609:        moveq     #%01101011,d0     * Wichtige Tasten
                                        ausmaskieren
610:        and.b     (a0),d0
611:*
612:        cmpi.b    #%1011,d0   * SHIFT-SHIFT-ALTERNATE?
613:        bne.s     MODE_END    * Nein...
614:*
615:        tas.b     KEY_FLAG    * War bereits gedrückt?
616:        bne.s     MODE_RTS    * Ja....
617:*
618:        move.b    MODE(pc),d0  * Modus wechseln
619:        subq.b    #1,d0        * (Nur möglich falls
                                     das Key-Flag
620:        bpl.s     SET_MODE     *  gelöscht war)
621:*
622:* gespeicherte Auflösung und Max-Wert für Mode holen
623:
624:        move.w    RESOLUTION(pc),d1   * Auflösung
625:        move.b    MAXMODE(pc,d1.w),d0 * Maximalwert
                                           je Auflösung
626:*
627:   SET_MODE:
        move.b    d0,MODE          * Neuen Modus setzen
628:        rts                        * Key-Flag bleibt
629:*
630:   MODE_END:
        clr.b     KEY_FLAG         * Flag löschen
631:*                                  * Tasten losgelassen
632:   MODE_RTS:
        rts
633:*
634:   MAXMODE:
        dc.b    1  * Max.Modus in niedriger Auflösung
635:        dc.b    1  * Max.Modus in mittlerer Auflösung
636:        dc.b    3  * Max.Modus in hoher Auflösung
637:        dc.b    0  * -
638:*
639:   ******************************************
640:*
641:   CONOUT:
        movem.l   a0-a2/d0-d2,-(sp) * Register retten
642:*
643:        move.l    28(sp),-(sp)       * Textadresse
644:        move.w    #PRINT_LINE,-(sp)
645:        trap      #GEMDOS
646:        addq.l    #6,sp
647:*
648:        movem.l   (sp)+,a0-a2/d0-d2 * Register laden
649:*
650:        move.l    (sp)+,(sp)  * Return-Adresse setzen
651:        rts
652:*
653:   ******************************************
654:*
655:        dc.l      $02101968          * Magic Number
656:*
657:   NEW_MOUSE:
        movem.l   d0-d3/a0-a1,-(sp) * Register retten
659:*
660:        move.b    (a0)+,d0     * Ist es ein Mauspaket?
661:        cmpi.b    #$f8,d0
662:        bcs.s     END_MOUSE          * Nein...
663:        cmpi.b    #$fc,d0
664:        bcc.s     END_MOUSE          * Nein...
665:*
666:        move.w    #319,d2            * Mausgrenzen je
                                           nach Auflösung
667:        move.w    #199,d3
668:        move.w    RESOLUTION(pc),d0  * Auflösung
669:        beq.s     CHK_MOUSE          * War niedere
670:*
671:        move.w    #639,d2
672:        subq.w    #1,d0
673:        beq.s     CHK_MOUSE          * War mittlere
674:*
675:        move.w    #399,d3            * War hohe
676:*
677:   CHK_MOUSE:
678:        move.b    (a0)+,d0           * Hole relative
                                           Mausbewegung x
679:        move.b    (a0)+,d1           * und y
680:*
681:        lea       MOUSE_POS(pc),a1   * Addiere
                                           Mausbewegung
682:        ext.w     d0    * auf Mauskoordinaten, prüfe
683:        ext.w     d1    * auf Bildschirmgrenzen
684:*
685:        add.w     (a1),d0      * xpos := xpos + rx
686:        bpl.s     MS_1
687:        moveq     #0,d0        * if xpos < 0,xpos = 0
688:*
689:   MS_1:
        cmp.w     d2,d0        * Abfrage anders
690:        bls.s     MS_2
691:        move.w    d2,d0        * if xpos > xmax,
                                     xpos = xmax
692:*
```

Listing geht weiter

```
693:  MS_2:
         add.w       2(a1),d1   * ypos := ypos + ry
694:     bpl.s       MS_3
695:     moveq       #0,d1      * if ypos < 0
                                  then ypos = 0
696:*
697:  MS_3:
         cmp.w       d3,d1      * Abfrage anders
698:     bls.s       MS_4
699:     move.w      d3,d1      * if ypos > ymax then
                                  ypos = ymax
700:*
701:  MS_4:
         move.w      d0,(a1)+   * Neue Mausposition
                                  sichern
702:     move.w      d1,(a1)+
703:*
704:  END_MOUSE:
705:     movem.l     (sp)+,d0-d3/a0-a1  * Register laden
706:*              * und weiter mit altem Mausinterrupt
707:     move.l      KBDVBASE+4(pc),-(sp)
708:     rts
709:*
710:  ******************************************
711:*
712:  CREATE_TABLE:
713:     lea         BUFFER_2,a0  * Tabelle für Faktor 2
                                    berechnen
714:     moveq       #0,d0
715:*
716:  MAKE_20:
         move.b      d0,d1
717:     moveq       #7,d7        * 8 Bits pro Byte
718:*
719:  MAKE_21:
         move.b      d1,d3        * Aus einem Byte
                                    ein Word machen
720:     add.b       d1,d1        * 2 x dasselbe Bit
                                    testen u. setzen
721:     addx.w      d2,d2
722:     add.b       d3,d3
723:     addx.w      d2,d2
724:     dbra        d7,MAKE_21
725:*
726:     move.w      d2,(a0)+
727:*
728:     addq.b      #1,d0
729:     bne.s       MAKE_20
730:*
731:  ********************************************
732:*
733:     lea         BUFFER_4,a0  * Tabelle berechnen
                                    (Faktor 4)
734:     moveq       #-16,d4      * Maske Nibble links
735:     moveq       #15,d5       * Maske Nibble rechts
736:     moveq       #0,d0        * Startbyte
737:*
738:  MAKE_40:
         move.b      d0,d1        * Aktuelles Byte in d1
                                    bearbeiten
739:     moveq       #3,d7        * 4 mal 2 Bits /Byte
740:*
741:  MAKE_41:
         add.b       d1,d1        * Mache aus 2 Bits
742:     scs         d2           * ein Byte
743:     and.b       d4,d2
744:     add.b       d1,d1
745:     scs         d3
746:     and.b       d5,d3
747:     or.b        d2,d3
748:     move.b      d3,(a0)+
749:     dbra        d7,MAKE_41
750:*
751:     addq.b      #1,d0        * Bis alle 256 Bytes
752:     bne.s       MAKE_40      * durchgearbeitet sind
753:*
754:  ******************************************
755:*
756:     lea         BUFFER_8,a0  * Tabelle für Faktor 8
757:     moveq       #0,d0        * Startbyte

758:*
759:  MAKE_80:
         move.b      d0,d1        * Aktuelles Byte in d1
                                    bearbeiten
760:     moveq       #7,d2        * 8 Bits pro Byte
761:*
762:  MAKE_81: add.b     d1,d1    * aus Bit Byte machen
763:     scs         (a0)+
764:     dbra        d2,MAKE_81
765:*
766:     addq.b      #1,d0        * Bis alle 256 Bytes
767:     bne.s       MAKE_80      * bearbeitet wurden
768:*
769:     rts
770:*
771:  *****************************************
772:*
773:  INSTAL:
         moveq       #3,d0        * Auflösung sichern
774:     and.b       $FFFF8260,d0
775:     move.w      d0,RESOLUTION
776:
777:     move.l      $70,VBL_VEC * VBL-Vektor umleiten
778:     move.l      #NEW_VBL,$70
779:*
780:     move.l      $B8,XBIOS_VEC * XBIOS umleiten
781:     move.l      #NEW_XBIOS,$B8
782:*
783:*
784:*
785:     move.l      $4F2,d0      * Pointer auf Sysbase
786:     beq.s       UNDEF        * Falls Null, dann
                                    direkt $E1B setzen
787:*          * (Zur Sicherheit falls Adresse = 0)
788:*
789:     move.l      d0,a0
790:     move.l      36(a0),a1    * Adresse des
                                  Kbdshift-Status
791:*                             * ab Tos-Version 1.2
792:*
793:     cmpi.w      #$0102,2(a0) * Tos-Version älter
                                    als 1.2?
794:     bcc.s       TOS_12       * Ja...
795:*
796:  UNDEF:
         move.w      #$E1B,a1     * Sonst Ram-Adresse
                                    direkt setzen
797:*
798:  TOS_12:
         move.l      a1,SHIFT_ADDR  * Adresse sichern
799:*
800:     rts
801:*
802:  ******************************************
803:*
804:  GRAF_MKSTATE:
805:     lea         CTRL_ARRAY,a0
806:*
807:     move.w      #79,(a0)+
808:     clr.w       (a0)+
809:     move.w      #5,(a0)+
810:     clr.l       (a0)
811:*
812:     subq.l      #6,a0
813:*
814:     move.l      #AES_DATA,d1  * graf_mkstate
815:     move.w      #$C8,d0
816:     trap        #2
817:*
818:     move.l      AES_DATA+12(pc),a0  * int_out
819:     move.l      2(a0),MOUSE_POS     * mouse x,y

820:*
821:     rts
822:*
823:  ******************************************
824:*
825:  APP_INIT:
         moveq       #10,d0
826:     bsr         AES_CALL
827:*
828:     rts
829:*
830:  ******************************************
831:*
832:  APP_EXIT:                   * Abmelden der
         moveq       #19,d0       * Applikation
833:     bsr         AES_CALL     * AES aufrufen
834:*
835:     rts
```

Listing geht weiter

```
836:*
837:   *****************************************
838:*
839:   AES_CALL:
           lea        CTRL_ARRAY,a0
840:*
841:       move.w     d0,(a0)+
842:       clr.w      (a0)+
843:       move.w     #1,(a0)+
844:       clr.l      (a0)
845:*
846:       subq.l     #6,a0
847:*
848:       move.l     #AES_DATA,d1
849:       move.w     #$C8,d0
850:       trap       #2
851:*
852:       rts
853:*
854:   *****************************************
855:*
856:   PAUSE:
           moveq      #30,d0         * Warteschleife
857:*
858:   WAITLOOP:
           dbra       d1,WAITLOOP    * Prozessor
                                       beschäftigen
859:       dbra       d0,WAITLOOP
860:*
861:       rts
862:*
863:   **********************************************
864:   *
865:   *          data
866:   *
867:   **********************************************
868:*
869:   KBDVBASE:   dc.l      -1   * Vektortabelle-Adresse
870:               dc.l      -1   * Gesicherter Maus-Vektor
871:*
872:   MOUSE_POS:  dc.w 639   * Initialdaten Mausposition
873:               dc.w 399
874:*
875:   FLIPBASE_1: dc.l -1        * Basisadresse Bild 1
876:   FLIPBASE_2: dc.l 32000     * Bild 2= Bild_1+32000
877:   PHYSBASE:   dc.l -1        * Hintergrund-Bildschirm
878:*
879:   XBIOS_VEC:  dc.l -1        * XBIOS-Adresse
880:   VBL_VEC:    dc.l -1        * VBL-Adresse
881:   SHIFT_ADDR: dc.l -1        * Tastaturstatus-Adresse
882:*
883:   RESOLUTION: dc.w -1        * Auflösung
884:   COUNTER:    dc.w REPEAT    * Countdown f.Bildaufbau
885:*
886:   AES_DATA:   dc.l CTRL_ARRAY
887:               dc.l AES_GLOBAL
888:                        dc.l CTRL_INT_IN
889:                        dc.l CTRL_INT_OUT
890:                        dc.l CTRL_ADDR_IN
891:                        dc.l CTRL_ADDR_OUT
892:*
893:   AES_GLOBAL: dc.w 0
894:                        dc.w 0
895:                        dc.w 0
896:                        dc.l $02101968
897:                        ds.l 5
898:*
899:   MODE:                dc.b 0  * Vergrösserung aus
900:   KEY_FLAG:            dc.b 0  * Flag "Tasten werden gedrückt
                                         gehalten"
901:*
902:   MESSAGE:             dc.b $1b,'E',10
903:                        dc.b ' R E A L T I M E
                                 M A G N I F I E R',10,10,10,13
904:                        dc.b '    Written by Alexander Beller'
                                ,10,10,13
905:                        dc.b '    (C) 1988 by IMAGIC Grafik',0
906:
907:                        ds.w 0
908:   TEXT_END:
909:
910:   ***********************************************
911:   *
912:             bss
913:   *
914:   ***********************************************
915:
916:   BSS_START:
917:
918:   BUFFER_2:        ds.b   256*2
919:   BUFFER_4:        ds.b   256*4   * Platz für Tabellen
920:   BUFFER_8:        ds.b   256*8
921:
922:   CTRL_ARRAY:      ds.w   10
923:   CTRL_INT_IN:     ds.w   128
924:   CTRL_INT_OUT:    ds.w   128
925:   CTRL_ADDR_IN:    ds.l   128
926:   CTRL_ADDR_OUT:   ds.l   128
927:
928:   FLIPSCR_1:       ds.b   32256   * Bildspeicher 1
929:   FLIPSCR_2:       ds.b   32000   * Bildspeicher 2
930:
931:   BSS_END:         ds.b   0       * Ende des residenten
                                          Teils
932:
933:                    ds.b   2048    * Programm-Stack
934:   USER_STACK:      ds.b   0
935:
936:   ***********************************************
937:
938:   end
```

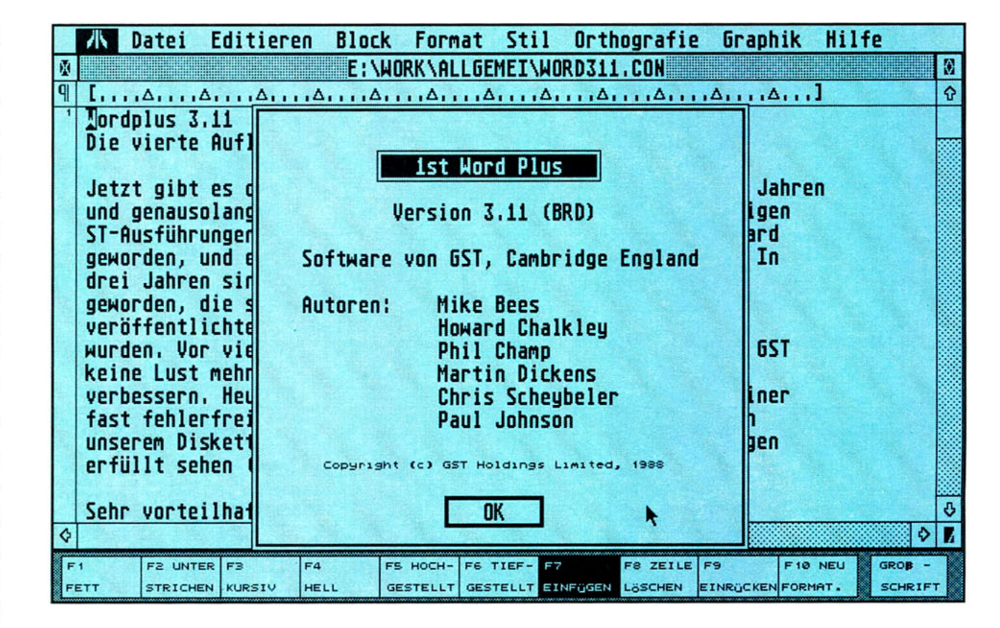
Listing der Lupe

ENDE

Wordplus 3.11

Die vierte Auflage von 1st_Word

Jetzt gibt es den ATARI ST schon seit ungefähr dreieinhalb Jahren und genausolang das “erste Wort”. Am Anfang, als es zu einigen ST-Ausführungen zum Lieferumfang gehörte, ist es zum Standard geworden, und es konnte seine Stellung bis heute bewahren. In drei Jahren sind aber einige Forderungen der Anwender laut geworden, die schon in der Version 2.02, der letzten veröffentlichten Version, nur sehr spärlich berücksichtigt wurden. Vor vier Wochen haben sich noch einige gefragt, ob GST keine Lust mehr hat, seinen Standard zu pflegen und zu verbessern. Heute haben wir die deutsche Version 3.11 in einer fast fehlerfreien Version (ATARI sagt Gammaversion dazu) in unserem Diskettenschacht und werden gleich unsere Forderungen erfüllt sehen (oder nicht?).



Sehr vorteilhaft beim Test einer Textverarbeitung ist die Möglichkeit seine Erfahrungen sofort niederschreiben zu können. Nach dem Laden fällt als erstes die Fileselektorbox auf. Nicht unbedingt positiv, da es die normale aus dem Betriebssystem ist, andererseits hat man so die Möglichkeit, eine beliebige aus der Public Domain-Sammlung zu wählen, die dann vielleicht genau das kann, was man unbedingt braucht, die aber auf jeden Fall besser ist als die des GEM.

Nachdem die zu schreibende Datei eröffnet ist und mit der Maus durch die Menüleiste gefahren wird, kommt zum ersten Mal Freude auf. In dieser Version haben von 65 Menüpunkten 54 hinter ihrer Funktionsbeschreibung noch einige zusätzliche Zeichen, die, wie ein kleiner Test auch gleich zeigt, die Tastenkombination beschreibt, die gleichfalls diese Funktion aufruft. Erstaunen kam nur wegen der Zuordnung der Buchstaben zu den Funktionen auf, denn sie haben keine Ähnlichkeit mit denen, die ich aus anderen Editoren gewohnt bin. Auch kann ich nur wenig Zusammenhang zwischen den Namen der Funktion (weder im Englischen noch im Deutschen) und der aufrufenden Taste entdecken, so daß eine kleine Hilfestellung gegeben wäre (*Control+C* sichert und schließt z.B. ein Dokument?!). Aber wie es nunmal ist: ein Standard schafft sich widerum seinen eigenen.

Soeben muß ich feststellen, daß kein Undo (zu deutsch so ähnlich wie rückgängig machen) möglich ist, so muß der obige Absatz, nachdem einige Zeichen versehentlich gelöscht wurden, von mir selbst umständlich restauriert werden. Bei diesem Text und an dieser eben erst geschriebenen Stelle ist es zwar nicht schlimm, doch der Rechner, hätte er eine *Undo*-Funktion, könnte es besser als ich.

Aber immerhin hat sich etwas bei den Blockoperationen getan. Ein Block wird nun nicht mehr mit einer aufzuziehenden Box markiert, sondern, wie in den meisten Programmen, die Blöcke unterstützen, wird mit gedrückter linker Maustaste über den zu markierenden Bereich gefahren. Die Darstellung als Block geschieht auch gleich, auf meinem Monitor in inverser Schrift. Mit dieser Art der Blockmarkierung ist es nun auch möglich, das Blockende und den Blockanfang beliebig auf dem Monitor zu plazieren. Sollte der zu markierende Block nicht auf eine Bildschirmseite passen, kann das Blockende bzw. der Blockanfang auch durch Doppelklick auf die gewünschte Position markiert werden. Leider sind immer noch keine spaltenorientierten Blöcke zu markieren, mit denen es dann möglich wäre, mehrspaltigen Satz zusammenzuflicken,

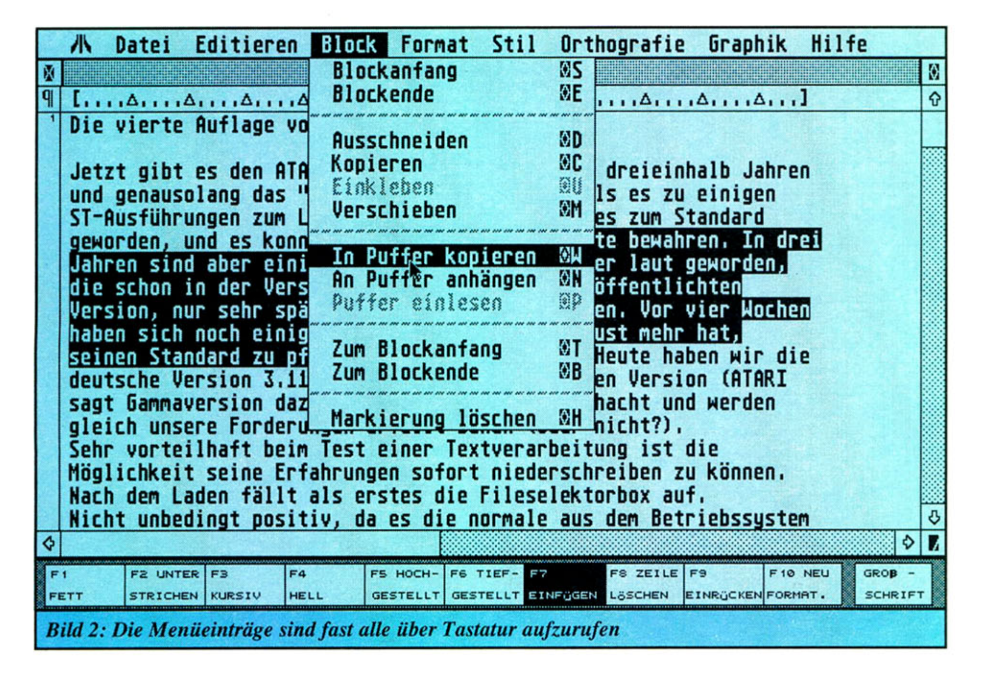


Bild 2: Die Menüeinträge sind fast alle über Tastatur aufzurufen

denn Mehrspaltensatz wird (ich bin fast geneigt, selbstverständlich zu sagen) nicht unterstützt. Ist ein Block markiert, und wird dabei auf kursiv oder eine andere Formatinformation geschaltet, wird der Block sofort dem gewünschten Aussehen angepaßt, ohne ihn wie bisher explizit "restylen" zu müssen. Als wirklich positiv ist der neu hinzugekommene Puffer zu werten. Zwar nicht unbedingt wegen der Ausführungsgeschwindigkeit, sondern vielmehr wegen den zusätzlichen Möglichkeiten, die er bietet. Ein markierter Block kann in Wordplus wie bisher behandelt (die Blockoperationen wurden also nicht ersetzt, sondern erweitert) oder in eben diesen Puffer kopiert werden. Dabei wird auf einen externen Speichermedium (sollte es ein Diskettenlaufwerk sein, dann ...) ein Ordner angelegt, in den der Block einmal mit Formatinformationen, also als WP-File, abgelegt wird, gleichzeitig auch ohne, mit der Extention TXT. So ist es endlich möglich, Blöcke mit allen nur denkbaren Schriftarten von einem Text in den anderen zu übernehmen. Als zweite gute Tat erlaubt dieser Puffer das "Mergen" (Anhängen) eines Blockes an einen [illegible] gespeicherten. Negativ aufgefallen ist, daß die Funktion "*Block löschen*" nicht mehr auftaucht. Stattdessen muß man nun den markierten Block ausschneiden und muß dies dann jedesmal bestätigen.

Die Hilfsdateien

Da wir nun sowieso schon die externen Dateien angesprochen haben können wir hier gleich weiter erzählen. An 1st_Word wurde immer kritisiert, daß es nicht möglich sei, eine geöffnete Datei zu drukken, bzw. eine gerade in Druck befindliche Datei zu öffnen. Wordplus umgeht nun ganz geschickt das Problem, indem es für jedes zu druckende Dokument zuerst eine extra Datei anlegt; eine bereits existente wird also erst in einen speziellen Ordner kopiert, die aktuelle Datei, sollte sie gedruckt werden, natürlich auch, um schließlich diese Kopie an den Drucker zu senden. Naja, Hauptsache, es funktioniert, wenn sich der Druckvorgang nur nicht mehr von einem heruntergeklappten Menü aufhalten lassen würde.

Die dritte Funktion, bei der externe Dateien angelegt werden, ist die Arbeit mit Hintergrunddatei. Diese Funktion erlaubt die Arbeit mit Dateien, die größer sind als der Arbeitsspeicher. Gerade nicht benötigte Teile des Dokumentes werden in eine Hintergrunddatei ausgelagert und bei Bedarf wieder zurückgeholt. Es ist zwar schön, diese Option wählen zu können, mir ist aber schleierhaft, wer bei den "überragenden" Geschwindigkeiten von 1st_Word mit so langen Texten am Stück arbeitet. Die Hoffung, daß die Programmierer von GST die Grafikroutinen des GEM auf Trapp bringen könnten, haben sich nämlich nicht erfüllt. Ist man die Geschwindigkeit anderer, zumeist Programmeditoren gewöhnt, die allerdings die GEM-Funktionen links liegenlassen, könnte von einer Drosselung der CPU auf 2MHz ausgegangen werden. Also mir ist auf jeden Fall fast keine Geschwindigkeitssteigerung aufgefallen, und ein Test der Scrollgeschwindigkeit schien mir recht sinnlos zu sein.

Die Details

Soweit zu den grundlegenden Änderungen. Was nun folgt, sind mehr oder weniger nur Detailverbesserungen. Daß im Grafikmodus nicht mehr in den kleinen Zeichensatz umgeschaltet wird, statt dessen die Proportionen für den Ausdruck richtig dargestellt werden, mag für den einen ein Vorteil sein. Für den anderen, der gerne mehr Zeilen auf den Bildschirm darstellt, ohne dabei etwas mit Bildern im Sinn zu haben, ist es eben keiner.

Bei der Gestaltung einer Druckseite wird der Anwender mit der neuen Version doch um einiges flexibler. So ist nun endlich die Wahl zwischen ein- und eineinhalbzeiligem Ausdruck möglich. Daß diese Einstellung nur für das gesamte Formular einzustellen ist, mag den einen oder anderen nicht ganz überzeugen können. Um dieses Manko auszugleichen, kann wenigstens bei der Erstellung eines neuen Lineals angegeben werden, ob, und wenn wieviele Zeilen übersprungen wer-

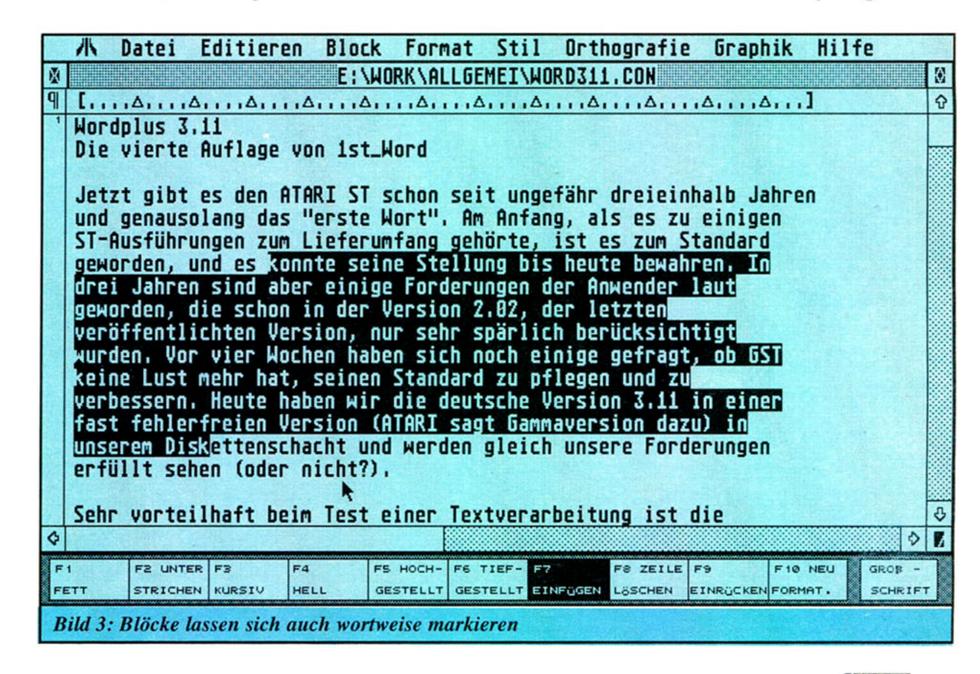


Bild 3: Blöcke lassen sich auch wortweise markieren

den sollen. Dies ist zwar nicht unbedingt eine feine Einstellung, doch lassen sich damit schon einige Effekte erzielen, die mit dem alten Wordplus nur umständlich zu realisieren waren. Auch ist die Wahl zwischen Blocksatz und Proportionalschrift möglich, wobei die proportionale Schrift nicht als solche auf dem Bildschirm dargestellt wird. Zum Glück möchte ich sagen, denn die Zeit, um auf die Ausgaben zu warten, hätte ich bestimmt nicht (Blocksatz kombiniert mit Proportionalschrift ist nicht möglich). Das Lineal wie auch die Positionsanzeige kann beliebig ein- und ausgeschaltet werden. Ebenso ist die Darstellung beider Anzeigen oder auch gar keiner möglich, wodurch eine Zeile mehr auf dem Monitor darstellbar wird. Eine weitere brauchbare Wahlmöglichkeit ist die, ob auf der ersten Druckseite Kopf- und Fußzeilen mit ausgegeben werden sollen oder nicht. Es ist also nicht mehr nötig sein Titelblatt, auf dem man sicher keinen Kopf benötigt, getrennt auszudrucken. Alles in allem haben gerade die wichtigen Druckersteuerkommandos eine gehörige Menge an Aufputschmittel erhalten, so daß ein differenzierterer Ausdruck als bisher möglich geworden ist.

Die langen Ladezeiten des Wörterbuches und die anschließende Dekomprimierung haben in der alten 2.02-Version die meisten Anwender abgehalten, diese Option zu nutzen. In der neuen ist bei den Ladezeiten eine deutliche Geschwindigkeitsteigerung zu beobachten (nur knappe 2 Minuten), und der erhebliche Zeitaufwand die Datei zu dekomprimieren, entfällt völlig. Daß ein Text nun auf Tastendruck auf Fehler bzw. auf unbekannte Worte hin untersucht werden kann, und

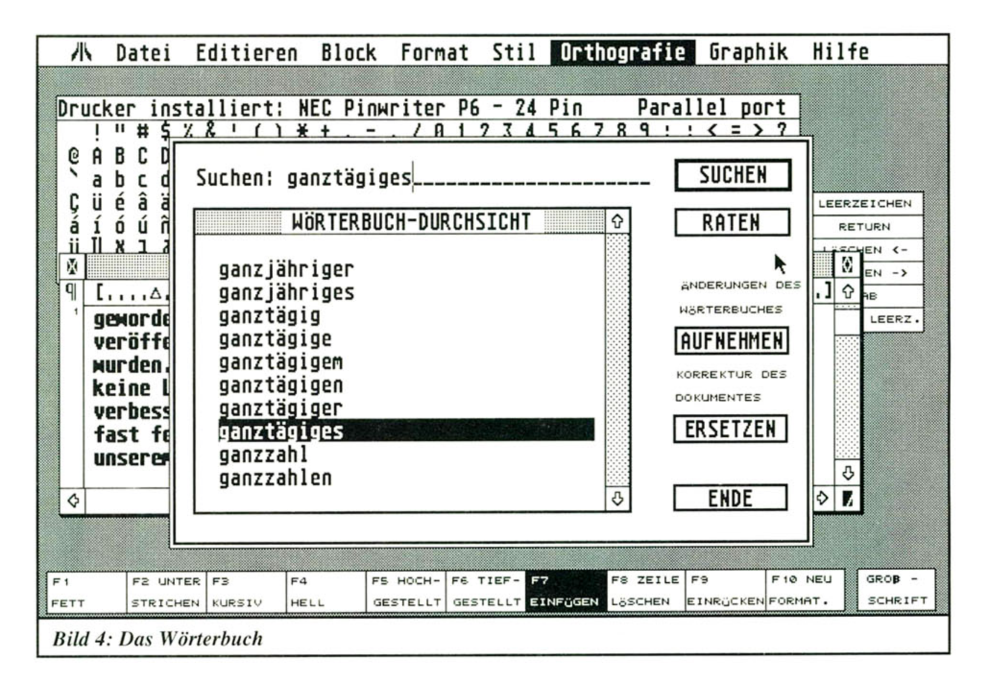


Bild 4: Das Wörterbuch

daß die so gefundenen Wörter mit einer Tastenkombination ins Wörterbuch zu übernehmen sind, macht diese Funktion doch erheblich attraktiver, und vielleicht greift jetzt auch der eine oder andere darauf zurück. Gleichfalls werden jetzt getrennte Worte als solche erkannt und nicht mehr als zwei unbekannte gewertet.

Über die Fußnotenverwaltung kann an dieser Stelle noch nicht viel gesagt werden, da sie nach Auskunft von ATARI-Deutschland zumindest noch einen groben Fehler enthält. Dieser wird allerdings bis zur Auslieferung korrigiert sein. Neu ist auf jeden Fall die mögliche Wahl eines Startoffsets. So kann der erste Fußnotenindex auf einen beliebigen Wert gesetzt werden.

Zum Schluß ein Schmankerl für zukünftige Anwender von Großbildschirmen, mit denen Wordplus, wie sich auf der ATARI-Messe gezeigt hat, hervorragend zusammenarbeitet. Nur die auf dem Bildschirm abgelegten Boxen wie die Funktionstastenleiste lagen auf diesem an ungünstiger Stelle. Das neue Wordplus umgeht das Problem ganz geschickt, indem die Boxen frei verschiebbar geworden sind. Es funktioniert natürlich auch auf allen anderen Monitoren, aber auf den kleinen sieht man bei geöffnetem Fenster sowieso nichts von ihnen. Diese Einstellung wie auch alle anderen, können nun in einer INF-Datei abgespeichert werden, wodurch sie bei jedem Laden automatisch eingestellt werden.

Fazit

Alles im allem hat 1st_Wordplus 3.11 doch eine erhebliche Aufwertung gegenüber seinem Vorgänger erhalten, auch wenn viele Wünsche nicht erfüllt wurden. Vor allem die Arbeit mit den Tastenkommandos macht sehr viel Spaß. Kleinere Fehler, wie teilweise oben erwähnt, wurden oder werden behoben. Daß die Ausgabe nicht sonderlich schneller geworden ist, liegt an der konsequenten Nutzung der GEM-Routinen mit allen Vor- und Nachteilen. Immerhin bietet Wordplus 3.11 einen deutlich größeren Leistungsumfang, ohne etwas am Konzept geändert zu haben. Und das erneute Studium des Handbuches ist auch nicht nötig, so daß der Austausch alt gegen neu ohne größere Probleme vonstatten geht. Von ATARI Deutschland wurde uns mitgeteilt, daß die neue Version 3.11 (entgegen anders lautender Meldungen) voraussichtlich erst nach der CeBIT '89 Ende März ausgeliefert werden wird.

WS

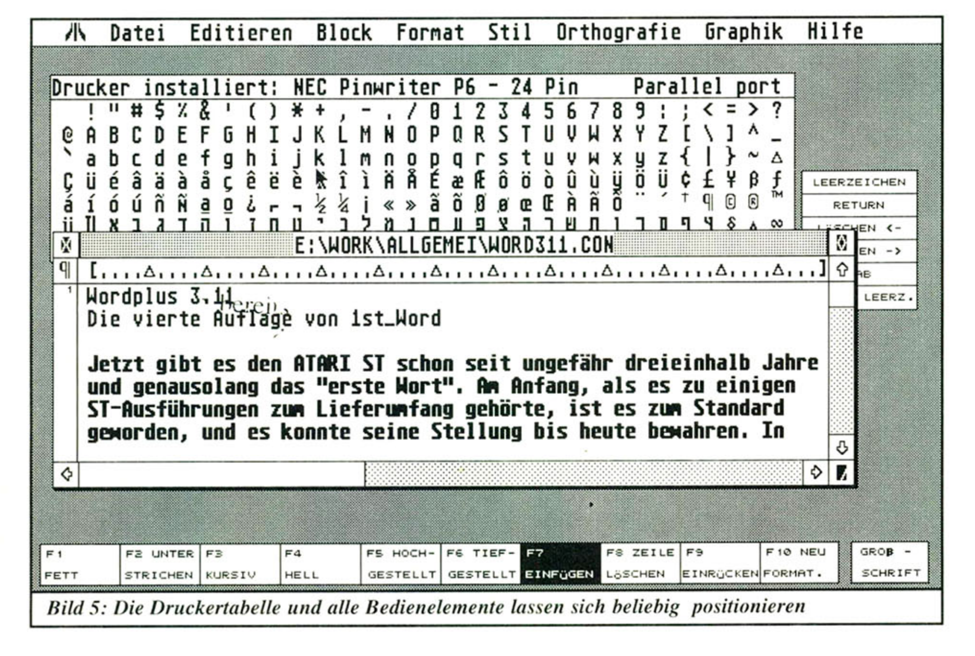
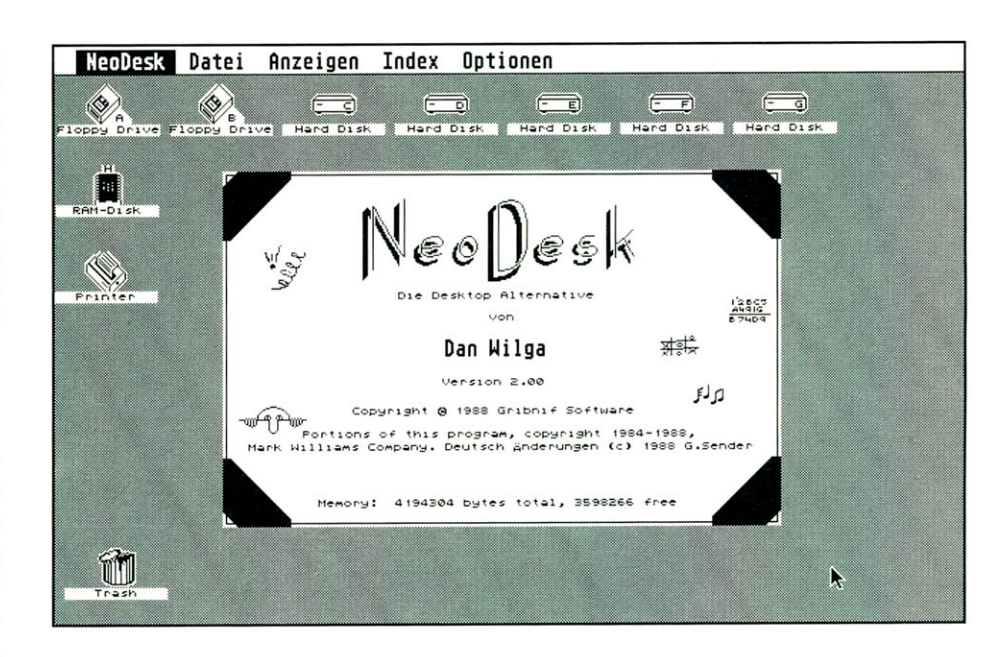


Bild 5: Die Druckertabelle und alle Bedienelemente lassen sich beliebig positionieren

NeoDesk

Das alternative Desktop

Vor einigen Monaten konnte man in den amerikanischen Mailboxen über ein neues Produkt namens NeoDesk lesen. Die Anwender überschlugen sich geradezu mit Lobhuldigungen zu diesem Programm. Auf der Düsseldorfer ATARI-Messe war es zum ersten Mal in einer Vorabversion in Deutschland zu sehen. Jetzt ist es ganz offiziell erhältlich.



Putzen wir die Platte

Vielen ist das "gute", alte Desktop inzwischen so ans Herz gewachsen, daß sie sich gar nicht mehr an Benutzeroberflächen anderer Computer erinnern wollen. Nun gut, andere können sich vielleicht unter dem Wort Desktop (Schreibtischoberfläche) überhaupt nichts Genaues vorstellen.

Folglich muß man zunächst einmal erwähnen, daß das ATARI-Desktop im Prinzip nichts anderes als ein Programm (eine sogenannte Shell) ist, das die tägliche Arbeit mit dem ATARI ST vereinfachen soll. Aus diesem Programm heraus kann man folglich einiges machen: z.B. Dateien verwalten, andere Programme starten usw. Es liegt also als äußere Schale um das Betriebssystems TOS (daher Shell) und bildet somit eine Schnittstelle zum Benutzer.

In der Neuen Welt machte sich nun ein Schreiner, pardon, Programmierer, daran, die Mängel des alten Desktops auszumerzen und schuf NeoDesk, eine alternative, aufpolierte Schreibtischoberfläche, die nur auf das alte Desktop draufgelegt werden muß, und schon häufen sich die neuen Möglichkeiten.

Es wird ausgepackt

Stellen wir zunächst einmal kurz die mitgelieferten Schreibtischutensilien und das Zubehör vor. In mehrere Ordner verteilt, findet man neben dem Hauptprogramm NeoDesk auch noch allerlei Nützliches.

Als Extras werden ein Commandline-Interpreter (für eingefleischte GEM-Hasser), ein Monitor für Speicher und Disketten (zum Wühlen in HEX-Codes u.ä.), ein Hardcopy-Programm für 24 Nadel-Drucker (alle Auflösungen und verschiedene Größen), das sich sowohl zum Ausdruck als auch zum Abspeichern auf Diskette eignet, ein Autostart-Programm für GEM-Anwendungen und ein Programm zum Stellen der Systemuhrzeit mitgeliefert.

Zwei Accessories sind ebenfalls zu finden. Bei ihnen handelt es sich um ein neues Kontrollfeld und eine Drucker-Warteschlange (einigen vielleicht besser unter dem neudeutschen Namen *Printer Queue* bekannt.).

Besser kontrolliert

Im NeoDesk-Kontrollfeld kann man im wesentlichen genau das machen, was man auch im ATARI-Original machen kann, also Uhrzeit und Datum, Wiederholungsrate der Tastatur usw. stellen. Leider wurde es nicht eingedeutscht, d.h. man hat eine amerikanische Schreibweise beim Datum (1.Dezember 1988 = 12/01/88 statt 01/12/88). Die Uhr läßt sich auf 12 oder 24 Stunden einstellen und in der rechten oberen Ecke der Menüleiste per-

manent anzeigen. Neu ist auch ein eingebauter Bildschirmschoner, der nach einer einstellbaren Zeit den Bildschirm invertiert, nicht schwarz schaltet! Ferner findet sich ein Schalter, um den Blitter (falls vorhanden) ein-/auszuschalten. Eine Anzeige des Gesamt- und des frei verfügbaren Speichers ist ebenfalls vorhanden.

Immer der Reihe nach

Will man mehrere Dateien nacheinander ausdrucken, bietet sich die NeoDesk-Printer Queue (tut mir leid, aber Drucker-Warteschlange klingt absolut übel) an. Hier lassen sich bis zu zehn Dateien auswählen, die nacheinader auf Papier gebracht werden sollen. Dank des Accessory-Characters ist dies auch aus GEM-Programmen möglich. Natürlich lassen sich Einträge für Leute, die kurz umdisponieren müssen, wieder aus der Warteliste löschen. Die Daten können auch an einen Druckerspooler übergeben werden. Sollte es mal nicht so klappen wie gewollt, kann man eine Druckerkonfigurationsdialogbox aufrufen, die der von ATARI mitgelieferten bis aufs I-Tüpfelchen gleicht. Ein deutlicher Pluspunkt ist, daß von den ausgewählten Dateien nur der Name und der Pfad abgespeichert werden, so daß nicht der gesamte Speicherplatz durch die auszudruckenden Dateien verbraucht wird.

Salve, großer Meister

Nach der Beschreibung des ganzen Zubehörs wenden wir uns jetzt dem eigentlichen NeoDesk-Programm zu, das auf den Namen *Neomaster* hört. Der Autor hat wirklich bei jeder normalen Desktopfunktion etwas verbessert, so daß wir nur auf die wichtigsten Funktionen kurz eingehen wollen.

Über die Menüleiste sind alle Funktionen des normalen Desktops zu erreichen. Daß allerdings noch wesentlich mehr zu finden ist als das Gewohnte, deutet schon die Infomeldung an, in der zusätzlich die Gesamtgröße des Arbeitsspeichers und der Umfang des frei verfügbaren RAM-Speichers angezeigt wird. Wer bis jetzt das Bild seiner Frau, Tochter, Freundin, Putzfrau (pardon, Mann, Sohn, Freund,...) auf dem Schreibtisch stehen hatte, der kann jetzt Platz sparen. Einfach irgendwo, irgendwie digitalisieren und anstatt des tristen, grauen (oder bei Farbe meist grünen) Desktophintergrunds laden. Eventuell ist man dann mit einem Bild seines Chefs mehr motiviert ("1984" läßt grüßen).

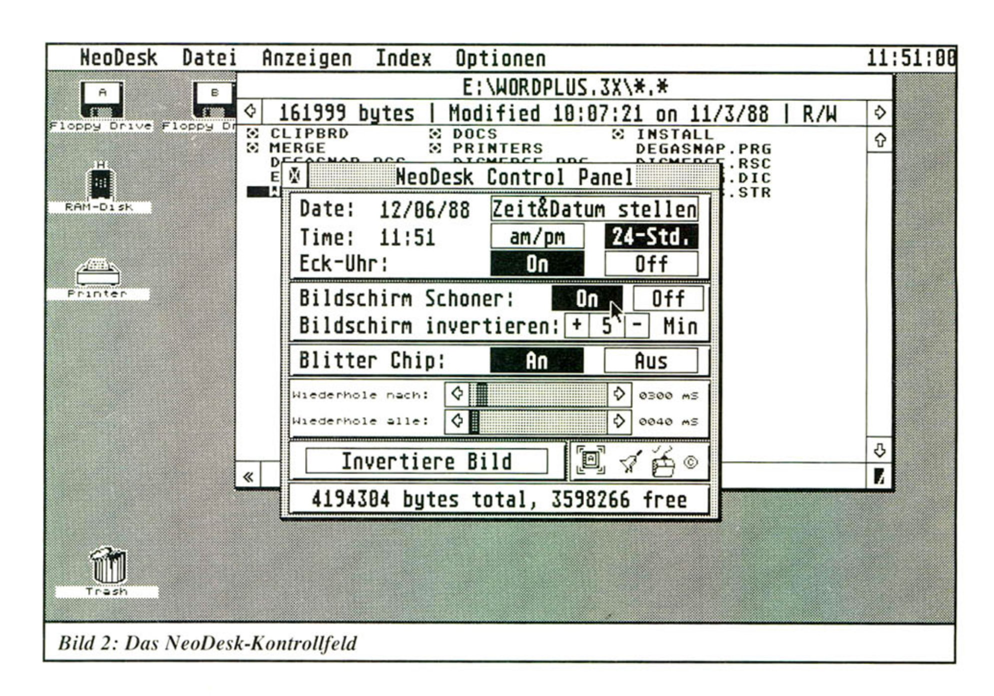


Bild 2: Das NeoDesk-Kontrollfeld

Tasten werden groß geschrieben

Zunächst sind alle Menüeinträge über Tastenkombinationen erreichbar, was einen deutlichen Gechwindigkeitsvorteil beim Arbeiten mit dem Computer bringt. Die entsprechenden Kombinationen findet man neben den Einträgen aufgeführt, so daß man sich langsam mit den ungewohnten Kürzeln vertraut machen kann. Ein bewährtes Prinzip, das leider nicht immer zu finden ist.

Nach gründlichem Studium des Handbuchs zeigt sich allerdings beim Umgang mit NeoDesk, daß immer wieder zusätzliche Funktionen durch Drücken der Tastenkombinationen zu erreichen sind. So lassen sich zum Beispiel alle Dateien anzeigen, egal ob Text-, Grafik- oder Programmdateien. Ein zweiter Anzeigemodus erlaubt das Vor- und Zurückblättern per Tastendruck.

Ungewohnte Fenster

Auch bei den Fenstern hat sich etwas getan. So lassen sich bis zu sieben Fenster gleichzeitig auf dem NeoDesktop öffnen, die Infozeile in einem Fenster scrollen und der Diskettenname anstatt des horizontalen Schiebers am unteren Ende anzeigen. Per Mausklick kann man Fenster hinter ein anderes versetzen und kann so schnell "umsortieren". Laufwerke lassen sich per Taste selektieren und öffnen.

Anzeige von Dateien

In einem Fenster lassen sich maximal 132 Dateinamen oder 72 Icons anzeigen. Dabei erfolgt die Anzeige wahlweise in einer Spalte oder mehreren. Beim letzte-

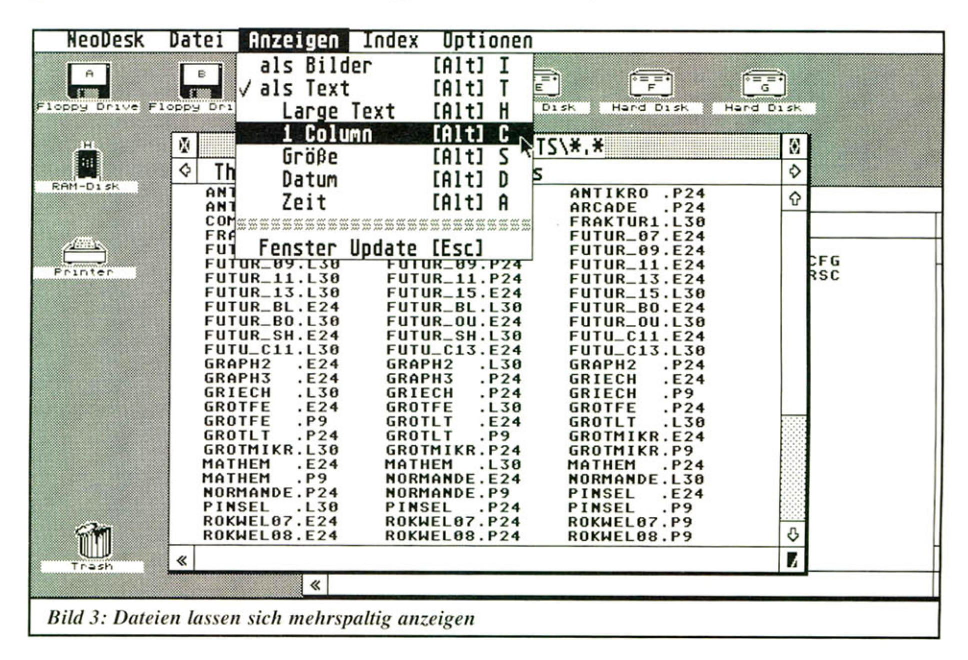


Bild 3: Dateien lassen sich mehrspaltig anzeigen

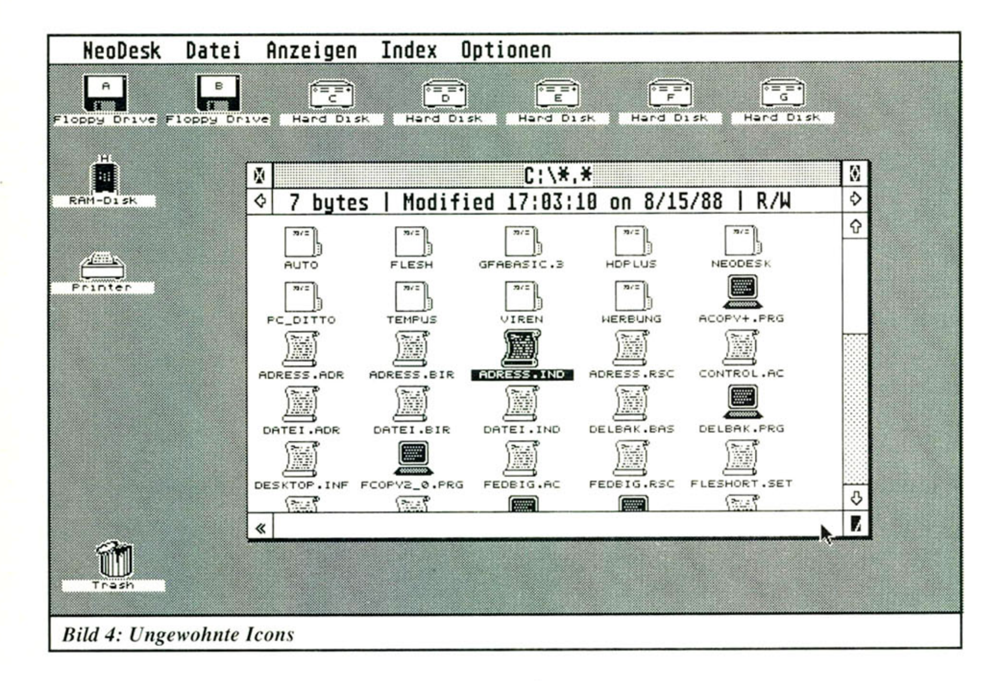


Bild 4: Ungewohnte Icons

ren wird von links nach rechts sortiert. Bei der Textdarstellung von Dateinamen kann man Größe, Datum und Zeit ausschalten, so daß man bei der Mehrspaltendarstellung mehr auf einmal sehen kann. Positiv wirkt sich dabei auch aus, daß man zwei Schriftgrößen zur Auswahl stehen. Zu den gewohnten Sortiermöglichkeiten ist eine unsortierte Anzeige hinzugekommen, die aber wohl kaum benutzt werden wird. Besser ist da schon die Filterung nach frei wählbaren Masken, wobei auch sogenannte Wildcards (? und *) benutzt werden können.

Auch beim Selektieren von Dateien sind große Variationsmöglichkeiten gegeben. Es lassen sich z.B. mehrere Textdateien selektieren und anschließend an eine Textverarbeitung oder Editor (z.B. Tempus) übergeben, worauf alle ausgewählten Dateien für sich getrennt geladen werden. Hat man eine Datei angewählt, erscheint in der Infozeile des betreffenden Fensters Größe, Erstellungsdatum und -zeit und der Status, ob eine Nur-Lesen- oder Lesen-und-Schreiben-Datei vorliegt. Selektiert man mehr als eine Datei, erscheint die Summe der Dateigrößen, was besonders zum Kopieren von Dateien sinnvoll ist, um zu sehen, ob genug Speicherplatz vorhanden ist.

Icons auf dem Desktop

NeoDesk bietet völlig neue Desktop-Icons. Vor allem sind noch einige bei der Gesamtanzahl hinzugekommen. So finden sich jetzt Drucker-, Festplatte-, Batchdatei- und RAM-Disk-Icons, die mit dem mitgelieferten Icon-Editor bearbeitet, gespeichert, geladen, getauscht usw. werden können. Allerdings können keine zusätzlichen Icons erzeugt, sondern nur die bestehenden modifiziert werden. Es besteht hier also ein Unterschied zum Macintosh oder AMIGA, wo jedes Programm ein Icon bekommt. Wenn wir gerade beim Mac sind: Es lassen sich Programme (als Icons) genau wie beim Mac direkt auf dem Desktop ablegen (max. 16), so daß man häufig benutzte Anwendungen ohne großes Geklicke in Fenstern leicht starten kann. Das so abgelegte Icon enthält allerdings nur Laufwerkskennung und Pfad, damit das Betriebssystem weiß, wo es das Programm suchen muß. Man findet es allerdings zusätzlich auch noch an gewohnter Stelle, von wo man es auf das Desktop gezogen hat.

Namenswechsel

Auch beim Ändern von Dateinamen tritt Erleichterung ein: einfach eine Taste und ein Mausklick - schon erscheint eine Dialogbox, in der der Name geändert werden kann. Will man mehrere Dateien umbenennen, selektiert man alle, und es erfolgt eine sequentielle Abarbeitung. Ebenfalls möglich ist das Anpassen von Dateien auf das aktuelle Systemdatum, was aber wohl ziemlich selten vorgenommen werden dürfte.

Auch Diskettennamen lassen sich nachträglich ändern. Dabei wird allerdings auf der betreffenden Diskette eine "Versteckte Datei" angelegt, die von NeoDesk dann immer wieder gefunden wird. Allerdings wird dieser Name dann ohne NeoDesk nicht gefunden. Der Nutzen ist also fraglich.

Hochformatig...

... geht es bei NeoDesk zu. Disketten lassen sich mit maximal 82 Spuren und 10 Sektoren formatieren, wobei MS-DOS-Lesbarkeit gewährleistet ist. Beim Kopieren von Disketten können wahlweise alle Dateien mit oder ohne Formatierung (automatisches Erkennen des Formats) dupliziert werden. Das Löschen ganzer Disketten erfolgt einfach durch Ziehen des Disketten-Icons auf den Papierkorb. Sehr praktisch!

Beim Punkt "Anwendung installieren" können auch Batchdateien ausgewählt werden ("Batchdatei übernimmt Parameter"). Schön ist auch die Möglichkeit, bis zu 10 sogenannte Environment-Strings einzugeben. Diese Strings sind im Prinzip Pfade, nach denen das Betriebssystem Dateien suchen kann, wenn es sie nicht an erwarteter Stelle gefunden hat. Einige

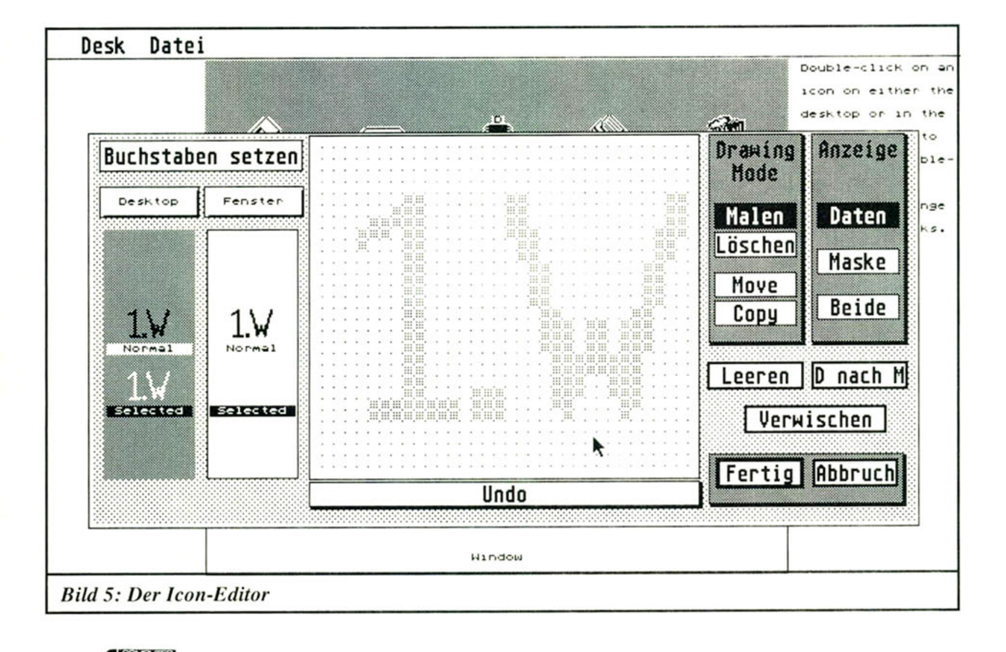


Bild 5: Der Icon-Editor

Anwender kennen vielleicht den MS-DOS-Befehl PATH, der im wesentlichen das gleiche bewirkt.

Voreinstellungen

Unter diesem Menüpunkt lassen sich vor allem Nachfragen zum Dateihandling aktivieren. So kann man z.B. entscheiden, ob eine Datei beim Ziehen in ein anderes Fenster kopiert oder verschoben (TOS 1.4 läßt grüßen) oder ob danach gefragt wird. Nützlich ist auch wieder die Funktion "Nach TOS- oder TTP-Applikationen anhalten". Dies braucht man bei der Programmentwicklung öfters, wenn z.B. der Compiler seine Fehlermeldungen ausgibt und, ehe man sich versieht, alles schon wieder hinter dem gewohnten Desktop verschwunden ist.

In den Voreinstellungen sind außerdem noch zusätzliche Pfade für das automatische Nachladen von Programmen nach dem Start von NeoDesk u.ä. zu bestimmen. Ebenso läßt sich hier die Anzahl der Dateien pro Ordner festlegen (Default = 112, Maximum = 999).

Fazit

Mit NeoDesk hat man mit Sicherheit einiges an Luxus beim täglichen Arbeiten mit dem ST gewonnen. Es stellt sich allerdings die Frage, ob man all diese zusätzlichen Funktionen auch wirklich braucht. Bei einem Preis von DM 89,- kann man sich aber nicht weiter beschweren, und sicher wird NeoDesk auch in Deutschland seinen Weg machen.

HE

Computerware
Gerd Sender
Weißer Str. 76
5000 Köln 50
Tel.: 0221/392583

ENDE

Der neue, kleine OMIKRON.-Assembler

Neu, weil er wenig mit dem alten Ideal-Assembler von OMIKRON gemein hat. Klein, weil demnächst eine Makroversion erscheinen wird. Die Makroversion wird im Gegensatz zu der vorliegenden neben Makros auch mit Objektmodulen arbeiten können und dazu natürlich auch einen Linker beinhalten. Der Assembler wird aller Voraussicht nach Objektcode im DRI- und auch im GST_Format erzeugen können. Diese Objektdateien kann dann selbstverständlich auch der Linker verarbeiten. Soviel zu dem, was der Kleine nicht kann, viel interessanter dürften aber seine Fähigkeiten sein.

```
OMIKRON. Assembler V1.02          Unser wichtigster Tester:
         von Σ-Soft                 Harald Weinreich
                                  Gute Tips & derbe Kritik:
Autoren: Sören Hellwig              Artur Södler
         Markus Fritze            Für die Centronics-Routinen:
Anleitung & RSC-Editor:             Thomas Hertzler
         Christoph Pagalies       Für die VDISK 3.0:
                                    Christian Markus
Seriennummer:8851                 Sonstigen Dank an:
MAXON Computer GmbH                 das gesamte OMIKRON.Team,
Industriestr. 26                    Darkstar, Rainer Zufall
6236 Eschborn                       sowie an alle, die uns kennen
```

Die Benutzeroberfläche kam mir auf dem ersten Blick sehr bekannt vor. In den oberen zwei Zeilen werden dabei die über die Funktionstasten oder Maus erreichbaren Funktionen dargestellt, der integrierte Editor setzt, genau wie das BASIC der gleichen Firma, die Eingaben sofort in tokenisierte Form um. Diese Vorübersetzung hat zur Folge, daß der reine Assemblerlauf auf Übersetzungsraten von durchschnittlich 1.1 Millionen Zeilen pro Minute kommt. Wenn man der Angabe, die nach jeder Übersetzung ausgegeben wird glauben darf - und ich habe keinen Grund dies nicht zu tun - wird diese Zahl im Normalfall noch um einiges übertroffen. Geht man aber von normal großen Sourcecode aus (bis 1000 Zeilen), würden 10% der Geschwindigkeit auch ausreichen. Der eigentliche Vorteil der Tokenisierung liegt neben dem geringeren Speicherbedarf auf Platte in der sofortigen Syntaxüberprüfung. Einige Fehler wie doppelte Vergabe eines Labelnamens werden so direkt bei der Eingabe abgefangen. Solange die als fehlerhaft erkannte Zeile nicht korregiert ist, weigert sich der Editor, diese zu verlassen bzw. denn weitere Zeilen anzunehmen. Die ausgegebene Meldung beschreibt den Fehler allerdings so genau, daß selbst Assemblereinsteiger, ohne lange in Büchern suchen zu müssen, die korregierte Zeile dem Assembler schmackhaft machen können.

Automatische Formatierung

Der dritte Vorteil schließlich ist die ordentliche Formatierung des Quelltextes (ohne eigenes Zutun). Da die Zeile nach der Eingabe erst einmal übersetzt, die tokenisierte Zeile dann wieder in ASCII-Text umgewandelt wird, der Assembler also sofort zwischen Labelnamen, Befehl und Kommentar unterscheiden kann, werden diese Teile bei der Bildschirmausgabe auf zuvor einstellbare Tabulatorpositionen gesetzt. Auch werden alle Befehle in Motorola-Standard umgesetzt, aus *move D0,A0* wird also *movea D0,A0*

Die Geschwindigkeit des Editors leidet unter dieser zusätzlichen Arbeit allerdings nicht. Die Arbeitsweise dürfte auch fast jedem ST-Anweder bekannt sein. Der neue ST-BASIC-Standard, OMIKRON.BASIC, arbeitet ja bekanntlich nach derselben Methode.

Da der Editor leider nicht in GEM oder wenigstens in eine nachempfundene GEM-Umgebung eingebunden ist, bleiben der Maus andere Aufgaben als die unter GEM üblichen wie Fenster verschieben oder in der Menüleiste herumfahren. Die auffälligste stellt dabei das Scrollen bei Erreichen des oberen beziehungsweise des unteren Bildschirmrandes dar (kann auch zu unschönen Erlebnissen führen, da der Rechner selbstverständlich nicht merkt, ob ein Scrollen auch beabsichtigt ist). Bei dauerhaftem Betätigen der linken Maustaste ist so das Markieren eines Blockes auch über die Bildschirmgrenzen hinaus möglich. Zumindest das ist eine Funktion, die ich jedem Editor "wünschen" würde.

Aber er bietet noch andere Besonderheiten, die in jedem Programmeditor möglich sein sollten. Für Assemblerprogrammierer, und hier besonders für den Einsteiger, stellen die oft benötigen Umrechnungen zwischen den verschiedenen Zahlensystemen ein Problem dar. Der OMIKRON.Assembler stellt dazu eine sehr gelungene und schnelle, den Arbeits-

```
Assembl Debug   Suche   Editor  Rechner Wdh.    Drucken Laden   Sichern 12:37:03
Accs            Ersetz. Info    Darst.  Löschen Spezial Zuladen A-Sich. Ende
Z:   47 Sp:  0  A:\DEMOS.OMI\AES_DEMO.SRC                           Überschreiben
                move.l  #$0B0000,D1
                bsr     vdi             ;v_opnvwk() - Workstation öffnen

;*****************************************************************************
;* Nun geht's mit dem Programm los                                            *
;*****************************************************************************

                lea     int_in(PC),A6   ;Zeiger auf int_in (Global!)
                bsr     init
                lea     wind_para(PC),A5 ;Window-Parameter (Global!)
mainloop:       lea     evnt_message(PC),A4
                movea.l A6,A0
                move.w  #$11,(A0)+      ;ev_mflags (Tastatur & Message)
                clr.l   (A0)+
                clr.l   (A0)+
                clr.l   (A0)+
                clr.l   (A0)+           ;Es wird nichts gebraucht
                clr.w   (A0)+
                clr.l   (A0)+
                clr.l   (A0)+
                clr.l   (A0)+
                move.l  A4,addr_in      ;evnt_message-Buffer
```

Bild 2: Eingabe beim Assembler

drang nicht bremsende Option zur Verfügung. Dazu wird der Cursor auf eine Zahl in beliebigen Format plaziert, durch Drücken einer bestimmten Tastenkombination wird diese Zahl der Reihe nach entweder in Hex-, Bin-,Dezimal- oder ASCII-Zeichen umgewandelt. Sollten diese einfache Umrechnungen nicht ausreichen, kann auch ein Rechner aufgerufen werden, der mit Zahlen in den oben genannten Zahlensystemen rechnen kann. Die so erzielten Ergebnisse können, sofern gewünscht, direkt in den Text übernommen werden.

Fehlerhandling

Im ganzen ist mit dem System sehr schön und vor allem sehr schnell zu arbeiten. Dazu gehört auch das schnelle Anspringen von fehlerhaften Zeilen, die erst beim Assemblieren entdeckt werden, und später, wenn der Text im ganzen übersetzt ist, das Austesten mit dem sehr leistungsfähigen Debugger.

Sollten beim Assemblieren Fehler gefunden werden (und das wird nicht ausbleiben), werden die fehlerhaften Zeilen intern mit Fehlermeldungen gespeichert. Nach dem Assemblerlauf kann dann auf Tastendruck von einer fehlerhaften Zeile direkt zur nächsten gesprungen werden. Dabei wird die jeweilige Fehlermeldung in der Statuszeile ausgegeben. Da es sich bei dieser Art von Fehlern zumeist um unbekannte Labels oder ähnliches handelt, die auf Schreibfehler zurückzuführen sind, dürfte die Erstellung eines lauffähigen Programmes nicht mehr allzu schwer fallen. Wenn einmal Texte, die mit anderen Assemblern erstellt wurden, also nicht nach den Spezifikationen von OMIKRON. tokenisiert sind oder sogar nicht die genauen Vorgaben des Motorola-Standards einhalten vorkommen, sollten auch keine Probleme entstehen, da auch ASCII-Dateien eingelesen werden können. Sollte im Quelltext allerdings die Unterstützung von mehreren Data- und Textsegmenten verlangt werden (kein Standard, aber in anderen Assemblern nicht unüblich), müssen die nicht tokenisierbaren Zeilen des Textes angepaßt werden. Dies gilt natürlich auch für Quelltexte von Assemblern, die sich eigene Pseudoopcodes geschaffen haben (Pseudoopcodes sind im Gegensatz zu Opcodes keine Befehle für den 68000-Prozessor, werden aber vom Assembler benötigt, um z.B. Speicher reservieren zu können. Dazu gibt es den Standardbefehl *DS.*). Von diesen selbstgeschaffenen Pseudoopcodes hat der OMIKRON.-Assembler allerdings auch einige zu bieten. Ich möchte auch nicht deren Nutzen für den Programmierer, der ständig mit seinem Assembler arbeitet, bestreiten, aber ich sehe schon die Programme für die Programmierpraxis, die von niemanden hier verstanden werden, weil massig unverständliche Opcodes Verwendung finden.

Assembler und BASIC

Jetzt stellt sich die Frage, für wen dieser Assembler am besten geeignet ist. Kleinere Programme, bei denen man schnell ein paar Hardwareregister beschreiben oder auf geschützte RAM-Bereiche zugreifen will, können sehr schnell geschrieben und ausgetestet werden. Große Programme, die man nicht in kleine Module zerlegen will, profitieren natürlich auch von den schnellen Übersetzungszeiten. Nicht geeignet ist er für Leute, die gerne Makros verwenden oder ihre Assemblerwerke in Hochsprachen einbinden wollen. Mit einer Ausnahme. OMIKRON.BASIC-Programmierer werden voll unterstützt, denn nachdem ein Programm assembliert ist, bleibt es dem Programmierer überlassen es als ausführbares Programm, als Datazeilen oder als String für den INLINE-Befehl des BASIC 3.00 abzulegen. In dieser BASICversion soll auch die Einbindung von relozierbaren, also an beliebige RAM-Adressen zu ladende Programme möglich sein. Das ist allerdings ein BASICproblem und soll hier nicht weiter interessieren. Es ist nur fraglich, ob sich

```
Assembl Debug   Suche   Editor  Rechner Wdh.    Drucken Laden   Sichern 12:44:01
Accs            Ersetz. Info    Darst.  Löschen Spezial Zuladen A-Sich. Ende
Z:   21 Sp:  0  A:\DEMOS.OMI\AES_DEMO.SRC                           Überschreiben
;*****************************************************************************
;* OMIKRON-AES-Demo (unvollendet)                                             *
;* ©1988 by OMIKRON. written by Markus Fritze         11.08.1988 15:05 *
;**************                                                      ******

                          Darstellung:

gem_init:   Befehlstab:  16     Opcodes      groß   klein           größe (1k)

            Operandentab: 24    Pseudoops    groß   klein

            Remarktab:   40     Register     groß   klein
                                                                    s
             Södler Tabs        Hexzahl      groß   klein
                                                                    dresse)
                 OK             Stack        A7     SP

              ABBRUCH           Zahl kürzen  ja     nein
                                                                    d (0)
            trap    #1              ;Mshrink(0,Basepageadr,Prglänge)
            lea     12(SP),SP       ;Nur noch den Stack korrigieren
;Das war schon alles. In A6 befindet sich noch die Basepageadresse und in a5
;liegt die totale Programmlänge (TRAP #1 zerstört D0-D2 und A0-A2!)
```

Bild 3: Parametereinstellungen

die von Haus aus schnellen OMIKRON.BASIC-Programme mit Assemblerroutinen noch ausschlaggebend beschleunigen lassen. Sollte einer der oben genannten Gründe auf Sie zutreffen, können Sie sich auch über den separaten Debugger freuen, der den Preis des Pakets eigentlich fast alleine rechtfertigen würde.

```
Trace   Do PC   Tracrts Ttraps  Skip PC Dir     Hexdump Disassm List     Switch
Tr68020 Tnosubs Tracrte Go      Overwrt Marker  Breakp  Info    Direct   Quit
  PC=09EC0C USP=0C5BF8 SSP=03594E SR=...10..... movea.l 4(SP),A6
◊ D0-D7 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
◊ A0-A7 00000000 00000000 00000000 00000000 00000000 00000000 00000000 000C5BF8
Start des Debuggers         :$013FF8
Ende des Debuggers          :$03FC5D
Start der freien Speichers  :$09EB00
Ende des freien Speichers   :$0C5C00
Start des TEXT-Segments     :$09EC0C  Länge:$00000648
Start des DATA-Segments     :$09F254  Länge:$00000092
Start des BSS-Segments      :$09F2E6  Länge:$000002DC
Symboltabelle               :69 Symbole
Erste freie Adresse         :$09F5C2
$09F344>l $09EC0C, $09EC0C+20
$09EC0C»!gem_init:          movea.l 4(SP),A6
$09EC10>!                   movea.w #$500,A5
$09EC14>!                   adda.l  $C(A6),A5
$09EC18>!                   adda.l  $14(A6),A5
$09EC1C>!                   adda.l  $1C(A6),A5
$09EC20>!                   move.l  A5,D1
$09EC22>!                   andi.w  #$FFFE,D1
$09EC26>!                   add.l   A6,D1
$09EC28>!                   movea.l D1,SP
$09EC2A>!                   move.l  A5,-(SP)
```

Bild 4: Der Debugger

Zusammenarbeit mit dem Debugger

Interessant ist in erster Linie die Zusammenarbeit mit dem Assembler. Der Debugger kann von diesen auf zwei Arten aufgerufen werden. Zum einen über eine Funktionstaste, zum anderen bietet der Assembler nach erfolgreicher Assemblierung die Wahl zwischen Abspeichern und der Übergabe des Programmes an den Debugger, wobei auch die Labelnamen übergeben werden können. Und schon kann man z.B. in Einzelschritten die Tätigkeit seines Programmes verfolgen. Sollte dabei festgestellt werden, daß eine Zeile geändert werden muß, gelangt man durch die Tastenkombination *Control+Help* in den Assembler zurück, und zwar genau an die Zeile, die zuletzt im Debugger ausgeführt wurde. Dieses schnelle Korrigieren führt mitunter, zumindest bei mir, zu recht sorglosem Umgang z.B. mit Schleifenzählern. Als ich noch mit einen recht langsamen Assembler gearbeitet habe, habe ich mir lieber eine Minute lang überlegt, wie oft eine Schleife durchlaufen werden soll, als meinen Source einmal mehr durch diverse Batchdateien zu quälen und dann den Fehler nur recht umständlich einkreisen zu können. Mit dem OMIKRON.-Assembler habe ich mir nun eine andere Arbeitsweise angewöhnt: *F1* für Assemblerstart drücken, der ist fertig, bevor ich den Finger von der Taste habe, dann *Return* drücken, und der Debugger meldet sich. Jetzt, z.B. um schnell an die gewünschte Stelle zu gelangen, "*g „schleife*" (steht für: Ausführen des Programmes bis zum Breakpoint Schleife), sie wird dann in Einzelschritten abgearbeitet. Ah.. der Wert stimmt nicht..., *Control+Help* - jetzt befinde ich mich wieder im Assembler, und zwar genau in der Zeile, die ich ändern möchte. Der ganze Vorgang hat nicht einmal eine Minute gedauert. Wozu soll ich mir also bei dieser Geschwindigkeit den Kopf wegen Kleinigkeiten wie Schleifenzählern zerbrechen.

Der Bildschirmaufbau des Debuggers entspricht den des Assemblers. In den oberen Zeilen sind also die Funktionen, die über die Funktionstasten erreichbar sind, aufgeführt. Darunter sind alle Register mit ihren momentanen Werten zu sehen, die durch Mausklick auch jederzeit mit anderen Werten belegt werden können. Die Maus kann hier ebenfalls zum Scrollen benutzt werden, darüber hinaus lassen sich durch Anwählen mit der rechten Maustaste beliebige Worte und Werte an die Cursorposition kopieren. Dadurch kann das lästige Abtippen von Adressen oder Sonstigem, was z.B. zum Setzen von Breakpoints benötigt werden, entfallen.

An Befehlsarmut krankt er auch nicht, so können über Funktionstasten schon vier verschiedene Arten des Tracens, also der Einzelschrittabarbeitung, gewählt werden. Soll das Tracen automatisiert werden, kann auch der UNTRACE-Befehl verwendet werden. Als Abbruch dienen hier Bedingungen, die sehr einfach mit *IF*-Anweisungen gesetzt werden. Eine Bedingung könnte z.B. *"IF ^d0=100"* lauten. Der UNTRACE-Befehl arbeitet dann entweder die angegebene Anzahl von Befehlen ab oder bricht den Testlauf ab, wenn das Register *D0* den Wert 100 angenommen hat. Dabei werden die letzten 256 ausgeführten Befehle und die dazugehörigen Registerinhalte im sogenannten Cachespeicher abgelegt, den man sich bei Bedarf auch ausgeben lassen kann. Auf diese Art lassen sich wirklich sehr einfach Fehler im Programm finden. Selbstverständlich sind auch alle Befehle zur "Speicherbearbeitung" wie das Anzeigen mit anschließender Änderung von beliebigen Speicherbereichen oder das Suchen nach bestimmten Speicherinhal-

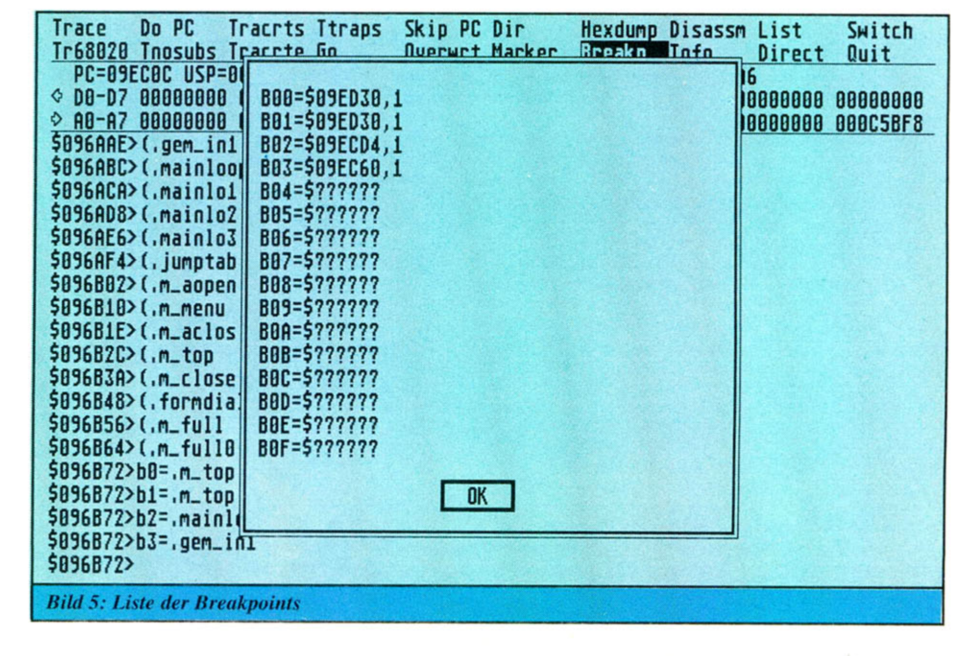


Bild 5: Liste der Breakpoints

ten implementiert. Dabei ist es völlig egal, ob der zu suchende Wert als Zahl, ASCII-Text oder als 68000-Befehl angegeben wird.

Einer der wichtigsten Befehle ist der *List*- bzw. der *Disassemble*-Befehl. Der unterschied zwischen den beiden ist, daß bei *Disassemble* zu den Mnemonic auch der Speicherinhalt ausgegeben wird, beim LISTen, sollte eine Symboltabelle mit dem Programm geladen worden sein, auch die Labelnamen. Sollte allerdings keine Symboltabelle geladen sein, erzeugt der Debugger leider keine eigenen Labelnamen, es ist also kein Reassembler. Dafür können alle Ausgaben sehr einfach auf einen Drucker oder auf eine Datei umgeleitet werden.
Um den Debugger auch jederzeit zur Hand zu haben, wird er durch den *Resident*-Befehl im Speicher gehalten, wobei er dann alle "Bombentraps" abfängt und die Suche der fehlerhaften Programmstelle ermöglicht. Der Aufruf des residenten Debuggers ohne Bombenwurf ist selbstverständlich auch möglich und erfolgt durch Betätigen von *Alternate+Help*, also der Kombination, die normal die Hardcopyfunktion aufruft. Sollte aber nicht der Debugger gemeint sein, kann zur Hardcopyroutine weitergesprungen werden.

Am Schuß möchte ich noch eine Besonderheit ansprechen, den *Switch*-Befehl. Daß damit zwischen Programmbildschirm und Bildschirm des Debuggers hin- und hergeschaltet werden kann, ist wohl nicht unbedingt etwas Besonderes. Sollte aber mit zwei Monitoren und einer Monitorumschaltbox mit Softwareumschaltung gearbeitet werden, schaltet der Debugger bei Erstellung von "farbigen" Programmen automatisch zwischen farbigem und monochromem Monitor um. Die Programmierer von OMIKRON. beweisen damit einmal mehr, mit welcher Liebe zum Detail sie arbeiten, und daß es auch an guten Ideen nicht fehlt.

Fazit

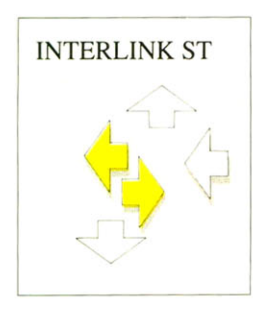
Der OMIKRON.-Assembler konnte, was die Bedienung und vor allem die Geschwindigkeit, mit der er assembliert, wodurch er eine sehr schnelle Programmentwicklung ermöglicht, einen sehr guten Eindruck hinterlassen. Daß die Erzeugung von Objektdateien und die Verwendung von Makrodefinitionen nicht möglich ist, mag den einen oder anderen ST-Anwender abschrecken, aber ich denke, es geht vielen wie mir, denn ich programmiere viel in Assembler, hatte aber noch nie das Verlangen, mit Makros zu arbeiten oder Assemblerroutinen in C einzubinden. Außerdem konnte nach kurzer Durchsicht der eingegangenen Karten unserer Leserumfrage festgestellt werden, daß schätzungsweise 70% unserer Leser in BASIC programmieren, die mit dem recht billigen OMIKRON.BASIC auch voll zufrieden sein werden. Sollte trotzdem jemand nicht auf diese Funktionen verzichten können und kurz vor dem Kauf eines Assemblers stehen, sollte auf das Erscheinen des großen, eben des Makroassemblers von OMIKRON. (noch im 1. Quartal '89) gewartet werden, es wird sich bestimmt lohnen. Die Preise betragen DM 99,- für den "kleinen" und DM 198,- für den Makroassembler.

OMIKRON.Software
Erlachstr. 15a
7534 Birkenfeld 2

ENDE

INTERLINK ST

Aller Kommunikationsanfang ist schwer. Doch wir möchten es Ihnen so leicht wie möglich machen. INTERLINK ST ist mit Sicherheit eines der komfortabelsten DFÜ-Programme für den ATARI ST, mit dem Sie weltweit kommunizieren können. Trotzdem überzeugt dieses Programm durch eine fast schon geniale Einfachheit. Zum Beispiel der eingebaute Recorder. Er funktioniert wie ein Cassettenrecorder. Einfach die "Taste" Aufnahme anklicken und schon wird die ganze Arbeit mit allen Einstellungen oder Texten, die Sie eingeben oder der angewählten Mailbox übertragen, mitgeschnitten. Und diese Aufnahme kann später wieder abgerufen werden. Dann führt INTERLINK ST die aufgezeichnente Kommunikation selber durch - zum Zusehen. Komfortabel, oder...?!



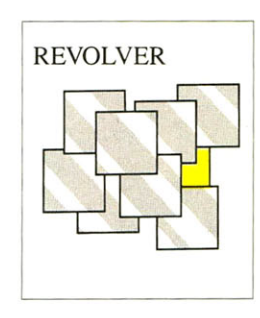
INTERLINK ST - DFÜ im Griff
Programm und Handbuch in Deutsch
Unverbindliche Preisempfehlung DM 79,-

REVOLVER

Bevor Sie jetzt in Deckung gehen, lesen Sie noch ein paar Zeilen weiter. REVOLVER ist kein Spiel und absolut ungefährlich für Sie! Nicht jedoch für Ihren ST, denn REVOLVER lauert im Hintergrund und handelt sofort. REVOLVER friert den Rechner ein! Jetzt kann der gesamte Inhalt des RAM-Speichers auf Diskette, Festplatte oder RAM-Disk abgespeichert werden. Zu jeder Abspeicherung sichert REVOLVER die zugehörigen Hardwareregister, alle normalen und residenten Programme, Deskaccessories usw. Die Speicherung erfolgt komprimiert. Die so eingefrorenen Programme können wieder geladen werden. Dann ist die Weiterarbeit an der gleichen Stelle möglich, an der das Programm ursprünglich unterbrochen wurde.

Um die Sache aber noch interessanter zu machen, teilt REVOLVER Ihren ATARI ST in Partitionen auf, wahlweise zwei bis acht Stück. Die Größe einer Partition kann in 256k Schritten eingestellt werden (z.B. 256k + 768k = 1 MByte). Jede Partition stellt einen eigenständigen ST dar und kann mit beliebigen eingefrorenen Programmen bestückt werden. Sie brauchen dann wirklich nur umschalten.

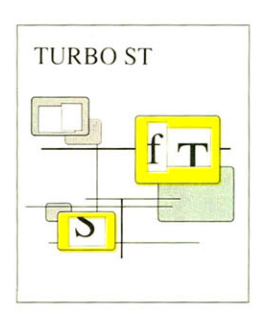
Doch REVOLVER kann noch mehr, z.B. umfangreiche Diskettenfunktionen wie Kopieren, Anhängen von Dateien, Umbenennen, Löschen, Verschieben (d.h. Kopieren und anschließendes Löschen an der ursprünglichen Stelle), Ordner erstellen und löschen. Befehlstrings können an alle Ports (also auch Keyboard oder Midiport) gesendet werden. Die Bewegungsgeschwindigkeit der Maus läßt sich im Faktor von 1:1 bis 1:6 variieren. Warm- oder Kaltstart sind ebenfalls möglich.



Mit REVOLVER wird das Ordnerproblem beseitigt. Die aufrufbare und beliebig große RAM-Disk liegt außerhalb der Partitionen und ist natürlich resetfest. Das gleiche gilt auch für den frei zu definierenden Druckerspooler.

REVOLVER - Der Volltreffer
Programm und Handbuch in Deutsch
Unverbindliche Preisempfehlung DM 129,-

TURBO ST



Das Programm Turbo ST ist die Softlösung des Blitters. Warum also noch auf die Hardware warten?! Besonders für Besitzer des 520 STM oder 1040 ST ist Turbo ST interessant. Denn Turbo ST ist **der** Softwareblitter!

Turbo ST wird als Desk-Accessory installiert und beschleunigt die Ausgabe von Texten auf dem Bildschirm. Dabei werden auf Text basierende Arbeiten sogar im Mega ST mit Blitterchip noch schneller. Textverarbeitung, Tabellenkalkulation, Programmeditoren oder einfach das Anzeigen von Texten im Desktop - die Ausgabe wird um ein vielfaches schneller gegenüber der normalen Anzeige.

Wie es funktioniert? Einmal installiert arbeitet Turbo ST, indem es Calls an GEM-DOS-Routinen abfängt und sie mit optimierten Routinen abarbeitet. Ohne Bastelei und löten. So einfach ist es, wenn was funktioniert!

Turbo ST - Der Softwareblitter
Programm und Handbuch in Deutsch
Unverbindliche Preisempfehlung DM 79,-

Vertrieb in Österreich:

Dipl.-Ing. Reinhart Temmel
Ges.m.b.H. & Co.KG
A-5440 Golling
Markt 109
Tel.: 06244/7081-17
Fax.: 06244/7188-3

Vertrieb in der Schweiz:

DTZ DataTrade AG
Langstrasse 94
Postfach 413
CH-8021 Zürich
Tel.: 01/242 80 88
Fax.: 01/291 05 07
(ausgenommen Turbo ST)

Vertrtieb in Holland:

JOTKA COMPUTING
Postbus 8183
NL-6710 AD Ede
Tel.: 08380/38731
Fax.: 08380/21675



BELA Computer Layout- und Vertriebs GmbH Unterortstraße 23-25 6236 Eschborn Tel.: 06196-481944 Fax.: 06196-481930

BILDWERKSTATT ATARI ST

Bildverarbeitung mit dem Computer

Im vorigen Teil dieser Serie haben wir Ihnen geschildert, wie Sie das Bildmaterial für einen Computerfilm erstellen können, wobei wir besonders ausführlich auf die Videodigitalisierung eingegangen sind. Diesmal kommen wir zu der vielleicht schönsten und zugleich auch kreativsten Phase unserer Arbeit: Wir zeigen Ihnen, wie der Film im Computer entsteht.



Die Bildermaschine

Jetzt setzen wir intensiv die Grafiksoftware IMAGIC ein, die speziell für die professionelle Erstellung von Animationen, Computerfilmen und selbstlaufenden Präsentationen konzipiert wurde. Das Besondere ist, daß IMAGIC mit einem sehr schnellen Zeichenprogramm, mit einer Bilddatenbank, die es erlaubt, bis zu 1000 Bilder gleichzeitig zu bearbeiten, und mit einem grafischen Filmeditor alle wichtigen Funktionen für die Erstellung von Computerfilmen in einem Programm vereint. Das Programm arbeitet zugleich in allen drei Auflösungen des ATARI ST.

Diejenigen unter Ihnen, die erst neu "eingestiegen" sind, möchten wir kurz darauf hinweisen, daß eine eingeschränkte Public-Domain-Version von IMAGIC parallel zu dieser Serie beim PD-Versand dieser Zeitschrift erhältlich ist. Damit können Sie zu Hause an Ihrem ATARI ST neben diesem Artikel die Schritte zur Erstellung von Computerfilmen mitvollziehen. Näheres dazu finden Sie in der PD-Übersicht in diesem Heft.

Wenn Sie die PD-Version von IMAGIC bereits zur Hand haben, bitten wir Sie um zwei Arbeitsschritte, die Sie vorweg durchführen müssen, damit Sie mit den Disketten arbeiten können. Beachten Sie bitte die Anweisungen in Tafel 1.

Damit Sie jedoch den Artikel nicht enttäuscht weglegen müssen, wenn Sie jetzt kein IMAGIC zur Hand haben: Wir versuchen, alle Arbeitsschritte möglichst allgemein verständlich zu beschreiben, so daß Sie auch bei einfachem Mitlesen einiges über die Erstellung von Computeranimationen erfahren können.

Von der Idee zur Animation

Ein Film aus dem Computer entsteht in mehreren Teilschritten, wir zeigen Ihnen die einzelnen Arbeitsschritte kurz auf:

Die Entwurfsphase

Hier werden zuerst die eigenen Ideen notiert, wird der Rahmen für den Ablauf abgesteckt. Bringen Sie Ihre Ideen zu Papier: Skizzieren Sie wichtige Teile aus dem Ablauf Ihres Computerfilms, versehen mit wichtigen Anmerkungen, und stellen Sie sich so ein kleines "Drehbuch" zusammen, das Ihnen hilft, die einzelnen Arbeitsschritte zu ordnen. In der Praxis, bei professionellen Computeranimationen, wird genauso vorgegangen: Dort nennt man ein solches Drehbuch *"Storyboard"*. Diese Arbeitsweise ist unumgänglich, wollen Sie einen Computerfilm aus mehreren Szenen erstellen und dabei die Übersicht behalten.

Im Filmeditor von DENISE, dem Animationsprogramm aus dem IMAGIC-Paket,

wurde das Storyboard deshalb in den Rechner verlegt: Für ein professionelles Erstellen von Computeranimationen enthält der Filmeditor ein elektronisches "Storyboard", mit dem der gesamte Ablauf des Computerfilms exakt anhand von Icons dargestellt und festgelegt werden kann. Gleichzeitig sorgt eine graphische "Bilddatenbank", in der alle Einzelbilder des Films verkleinert dargestellt sind, für die notwendige Übersicht bei der großen Menge an Einzelbildern, die ein Computerfilm benötigt. Eine Besonderheit von IMAGIC ist die Möglichkeit, das Storyboard immer sofort "abzuspielen", um einen ersten Eindruck vom Ergebnis zu erhalten.

Die Entstehungsphase

Jetzt werden alle Einzelteile einer Computeranimation bis ins Detail genau Bild für Bild erstellt. Dabei gewinnt man das erforderliche Bildmaterial entweder durch Videodigitalisierung und Scannen, wie wir es bereits geschildert haben, oder durch Berechnung im Computer, beispielsweise durch ein RAYTRACE- oder 3D- Programm. Weitere Programme sind speziell dafür gedacht, Bewegungsabläufe in Einzelbildern oder Trickfilmsequenzen zu erzeugen. Dazu zählen sowohl die Programme CYBER PAINT als auch der in diesen Tagen erscheinende CREATOR.

In IMAGIC setzen Sie den Zeichenteil von DENISE ein, um die Einzelbilder eines Computerfilms zu überarbeiten oder zu erstellen. Spezielle Zeichenfunktionen in DENISE sind besonders für die halbautomatische Erstellung von Computerfilmen oder Computeranimationen gedacht:

Beispielsweise das Arbeiten über angrenzende Bildschirme hinweg, um einen beliebig großen Hintergrund für eine Animationssequenz zu erstellen, oder das Zeichnen, Sprühen und Füllen mit Bildausschnitten, um Collagen im Computer zu ermöglichen oder Bilder auf zweidimensionale Oberflächen zu legen, oder das Durchkopieren von Bildelementen in andere Bilder nach dem BLUEBOX-Verfahren, um einen Zeichentrickfilm zu erstellen.

Das schnelle, gradweise Drehen und exakte Vergrößern /Verkleinern von Bildausschnitten kann zur Erzeugung von Zoomeffekten und Kamerafahrten im Computer eingesetzt werden.

Für die Weiterverarbeitung von Computergrafiken aus anderen Programmen ist DENISE in der Lage, alle unkomprimierten und die gängigen komprimierten Grafikformate am ATARI ST einzulesen. Laderoutinen für die Formate der amerikanischen ANTIC-Software CAD-3D und CYBER PAINT sowie für Grafikformate des APPLE MACINTOSH sind in Vorbereitung.

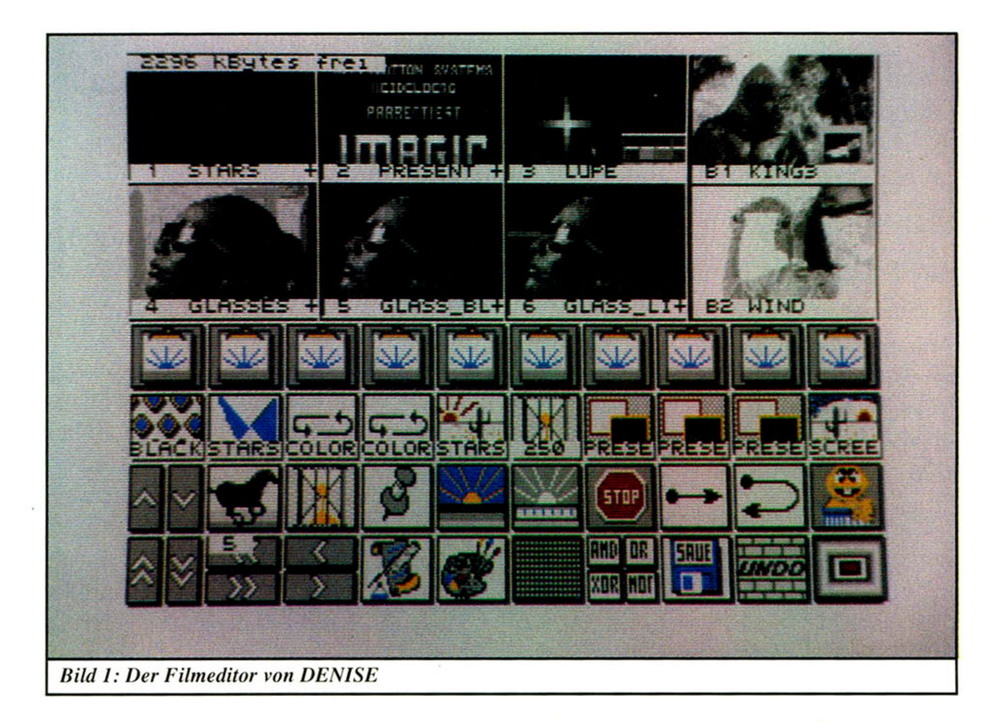


Bild 1: Der Filmeditor von DENISE

Die Bearbeitungsphase

Wie ein großes Puzzle setzen Sie jetzt die einzelnen Teile einer Animation zusammen. Diesen Teil der gesamten Arbeit möchten wir Ihnen gleich genauer vorstellen.

In unserem Beispiel erstellen wir einen Computerfilm mit mehreren Szenen und einer Gesamtlaufzeit von immerhin 3,5 Minuten, der komplett aus dem Speicher des Rechners ablaufen soll. In der Konzeption wird der Film in sieben Teile zerlegt, die in etwa den einzelnen Szenen entsprechen. Alle Teile werden einzeln bearbeitet und erstellt. Ist alles fertig, werden die einzelnen Teile zum Schluß zusammengefügt.

Die Darstellungsphase

Der fertige Computerfilm wird anschließend auf Diskette oder Videoband gespeichert, je nachdem ob er später aus dem Computer heraus oder von einem Videoband laufen soll. Zum Abspielen aus dem Computer heraus gibt es in den meisten Fällen ein sogenanntes "Run-Time-Modul", das ähnlich einem Run-Only für BASIC zusammen mit dem fertigen Computerfilm auf Diskette gespeichert

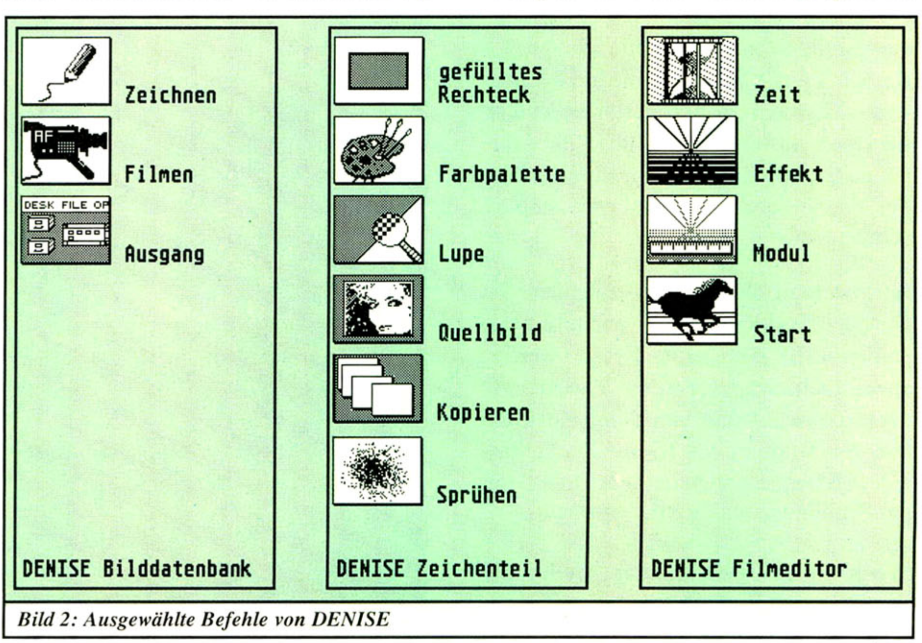


Bild 2: Ausgewählte Befehle von DENISE

wird. Für die Aufzeichnung auf Video setzen Sie einen Modulator oder ein GENLOCK-Interface ein, wie im vorigen Teil dieser Serie bereits geschildert.

Am Anfang war das Nichts ...

... genauer gesagt: Wir beginnen unsere Animation erst einmal mit einem schwarzen Bildschirm, langsam beginnen darin Sterne zu blinken, und dann schieben sich überdimensionale Buchstaben in das Bild.

Gesagt - getan: Starten Sie das Programm DENISDEM.PRG von Diskette 1! Quittieren Sie die Anfangsmeldung mit der Taste <RETURN> und warten Sie, bis die DENISE-Bilddatenbank erscheint. Klikken Sie jetzt auf das Icon *"Zeichnen"*, so befinden Sie sich im Zeichenteil von DENISE.

Als erstes benötigen Sie ein Bild mit schwarzem Hintergrund, in dem sich einige kleine Sterne befinden. Wenn Sie in Farbe arbeiten, können Sie die Sterne sogar parallel zu einer laufenden Animation blinken lassen, indem Sie die Technik der Farbanimation einsetzen. Dabei werden ausgewählte Farbregister des ATARI ST rhythmisch miteinander vertauscht, bei geschickter Wahl der beteiligten Farbregister kann daraus ein Bewegungs- oder Blinkeffekt erzeugt werden. Arbeiten Sie an einem Schwarzweißmonitor, läßt sich diese Technik nicht einsetzen.

Klicken Sie mit der Maus auf das Icon *"gefülltes Rechteck"*. Wenn Sie sonst nichts verändert haben, sollte ab Programmstart die Zeichenfarbe *"Schwarz"* eingestellt sein. Ziehen Sie ein großes Rechteck auf, klicken Sie die linke Maustaste. Mit dem so definierten Rechteck streichen Sie den Bildschirm ab, dabei halten Sie die linke Maustaste gedrückt, bis der gesamte Bildschirm komplett schwarz ist.

Jetzt werden die Sterne gezeichnet. Sie bestehen aus kleinen Kreuzen und einzelnen Punkten. Die kleinen Kreuze werden am einfachsten in mehreren Varianten in der Lupe erstellt und dann durch Kopieren über das Bild verteilt. Kleine Punkte lassen sich bequem in beliebiger Dichte mit der Sprühdose in das Bild "streuen".

Wenn Sie in Farbe arbeiten, stellen Sie sich jetzt eine Farbpalette für die blinkenden Sterne zusammen: Mit einem Mausklick rechts schalten Sie um in das Auswahlmenü von DENISE. Klicken Sie sodann links auf das Icon *"Farbpalette"*. Sie befinden sich in der Farbauswahl, mit einer zweizeiligen Darstellung der aktuellen Farbpalette im oberen Bildteil sowie einer Farbtafel, die jeweils 64 aus den 512 möglichen Farben des ST darstellt. Stellen Sie als erstes die Randfarbe auf schwarz: Klicken Sie auf das erste Feld der Farbtafel oben links (schwarz) und schieben es in das erste Register der Farbpalette oben links. Mit einem Mausklick wird die Farbe abgelegt. Stellen Sie sich jetzt eine Farbpalette mit 5-6 fein abgestuften Blautönen zusammen. Den Blauanteil der Farbtafel stellen Sie ein, indem Sie mit der Maus in den Streifen mit den Zahlen 0..7 links klicken. Schieben Sie die ausgewählten Farbwerte in die Farbpalette und legen Sie sie dort ab. Verlassen Sie die Farbauswahl über den Button *"EXIT"*.

Zurück im Auswahlmenü, klicken Sie mit der rechten Maustaste, dann gelangen Sie wieder in das Zeichenmenü von DENISE. Klicken Sie auf das Icon *"Lupe"*. Ein kleines Rechteck zeigt jetzt den Bildausschnitt an, den Sie bearbeiten können. Mit einem weiteren Mausklick links können Sie in der Lupe kleine Sterne zeichnen. Die Zeichenfarbe können Sie dabei direkt in der Lupe anwählen. Über die Funktionstasten F1 bis F10 stellen Sie die Vergrößerung der Lupe ein. Wenn Sie in Farbe arbeiten, versehen Sie zusätzlich die Sterne mit feinen Abstufungen in

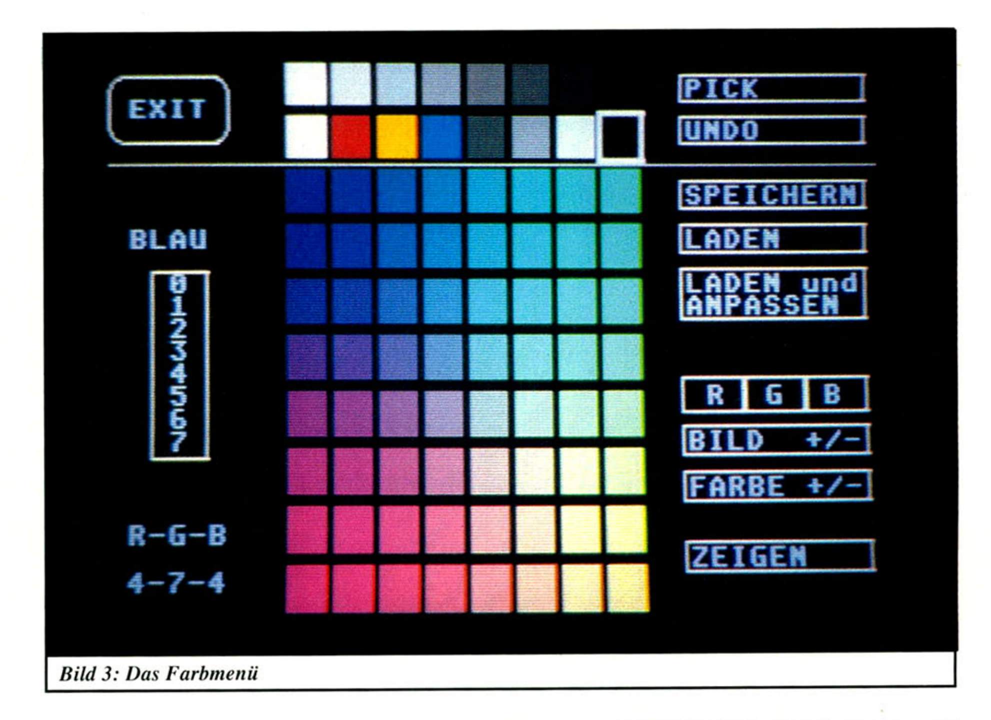


Bild 3: Das Farbmenü

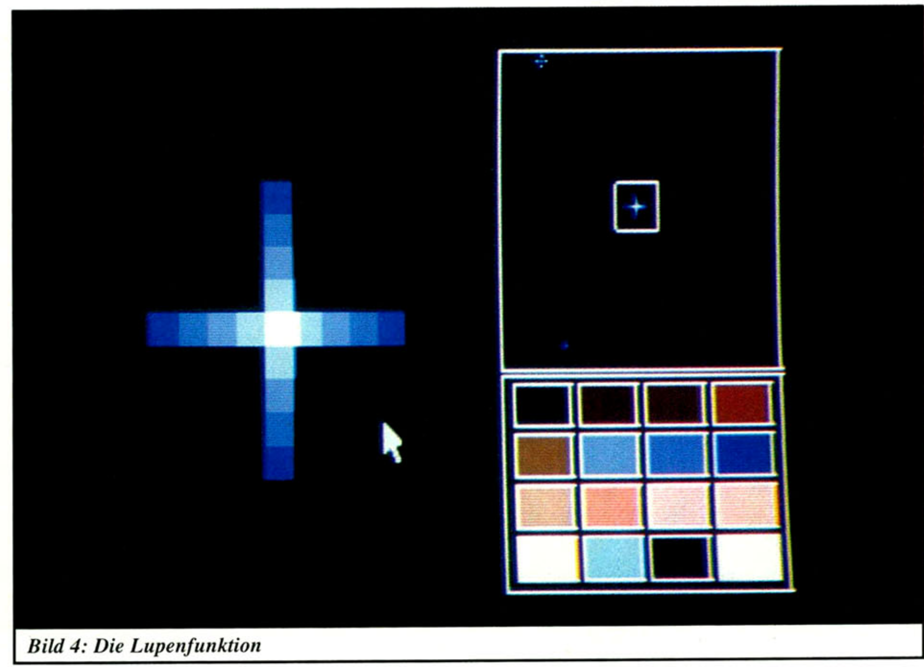


Bild 4: Die Lupenfunktion

WIR MACHEN MUSIK, DA GEHT EUCH DER BART AB.



Alle mal herhören: Die neue ST-Praxis spielt auf, als ob Jack Tramiel die Jukebox erfunden hätte. Ob klassisch oder schrill, wer einen Atari ST hat, der besitzt auch schon den Grundstock zum eigenen Musikstudio.

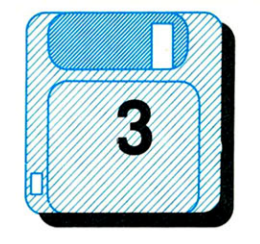
Unmusikalisch? Bei TOS kann sich kein ST-Freund taub stellen. Das neue Betriebssystem wurde monatelang heimlich in Deutschland getestet. Die ST-Praxis zeigt, wie es aussieht, und verrät erstmals alle Stärken und Schwächen der brandaktuellen TOS-Version.

Da klingen die Ohren: Als erste Computerzeitschrift bietet die ST-Praxis ihren Lesern eine Mailbox an, aus der alle im Heft abgedruckten Listings per Telefon abgerufen werden können. Kein mühsames Abtippen von Hand mehr - und natürlich für Anfänger ein Grundkurs in Sachen DFÜ (Datenfernübertragung).

Virtuose Tastentricks: Dreißig raffinierte Lösungen für oft vorkommende Einsteiger-Probleme und viele weitere Tips und Kniffe zu den wichtigsten Programmen - von der Textverarbeitung bis zum Computer Aided Design.

Dazu Marktübersichten, ein Programmierworkshop in Modula, ST-Tuning (Fremdmonitore am ST), Literaturtips, eine Spielebox und vieles mehr.

Mit Musik geht alles besser: ST-Praxis. Jetzt wieder neu am Kiosk.



STARKE SOFTWARE

Neue Version

ST LEARN
DER VOKABELTRAINER FÜR DEN ST

programmiert v. D Owerfeldt, Gewinner der GOLDENEN DISKETTE '87 für das beste Lernprogramm.
- Läuft vollständig unter GEM
- „Intelligente" Auswertung der Benutzereingaben
- Fehlerhäufigkeit einer Vokabel wird berücksichtigt
- Berücksichtigung mehrerer Bedeutungen eines Wortes
- Vielfältige Möglichkeiten des Lernens und der Abfrage
- Integriertes Lernspiel „HANGMAN"
- Spezielle Auswertung für unregelmäßige Verben (bei Eingabe von „to go" werden die anderen 2 Formen nachgefragt)
- Bei offensichtlicher Ähnlichkeit der Wörter wird wahlweise ein 2. Versuch zugelassen
- Trotz Einordnung der Vokabeln nach Lektionen oder Wissensgebieten ständig schneller Zugriff auf **alle** Vokabeln (nur durch Größe des Speichermediums begrenzt)
- Voller europäischer Zeichensatz (Zugriff durch die Maus unter GEM)
- Auch für Farbmonitor in mittlerer Auflösung
- Wörterbuchfunktion durchsucht alle Files eines Speichermediums nach einer Übersetzung ab
- Verbessertes Eingabeformular

ACHTUNG!!! NEU an ST-LEARN:
Allgemeines – Alle GEM-Routinen wurden überarbeitet u. optimiert, das Programm ist noch schneller geworden. – Besitzer eines SW-Monitors kommen in den Genuß einer neuen, etwas unkonventionellen (C) Mitteilung unter dem Menüpunkt **Über ST-LEARN.** – In der unteren Sonderzeichenleiste sind weitere Sonderzeichen hinzugekommen. – Im Lieferumfang befinden sich **jetzt auch mehrere englische Vokabeldateien, mit über 1.600 einfachen Grundwortschatzvokabeln,** unterteilt in 2 Schwierigkeitsstufen:
***EASY1.VOK** und **EASY2.VOK** enthalten je 400 einfachere Vokabeln.
***HARD1.VOK** und **HARD2.VOK** enthalten 800 etwas schwerere Vokabeln.
Weiterhin werden noch einige kleinere Vokabeldateien mit wichtigen Wörtern, etwa Strukturwörter oder Ordnungswörter mitgeliefert.
Eingabe- und Editierfeld – Hier ist der Knopf **'Löschen'** hinzugekommen, der es ermöglicht, die gerade angezeigte Vokabel zu löschen. Alle nachfolgenden Vokabeln werden aufgerückt.
Auswertung – Die Auswertung der Vokabeln wurde weiter verbessert, sie ist noch differenzierter geworden.
Druckeranpassung – Es ist jetzt möglich, **ST-LEARN** auf wirklich **jedem Drucker** anzupassen. Dazu gibt es im Ordner **LEARN_IT** das Konfigurationsprogramm PRINTER.
Einschaltmeldung – Hinzugekommen ist eine Einschaltmeldung, die direkt nach dem Starten des Programmes erscheint. Sie zeigt den freien Speicherplatz an.
Weitere Vorzüge von ST-LEARN – Vokabeldateien können auch nach Themen angelegt werden. Dennoch ist der Zugriff auf **alle** auf Diskette vorhandenen Vokabeln möglich. – Der **Status** einer Vokabel drückt deren Bekanntheitsgrad aus. Ist der Status z. B. kleiner als Null, so wurde dieser Ausdruck mindestens einmal nicht gekonnt etc. Je niedriger der Status, desto schlechter wurde die Vokabel nicht gekonnt und desto öfter wird sie vom Programm abgefragt.

DM 69,–

Update **DM 19,–**



ST DIGITAL
LOGIKSIMULATOR FÜR DEN ATARI ST

Ein Programm zum Erstellen, Testen und Analysieren von Logikschaltungen für Ausbildung und Hobby-Elektronik.
- Komfortable GEM-Umgebung
- Bauteile lassen sich per Maus plazieren u. verdrahten
- Umfassendes Bauteile-Set (Grundgatter, Ein-/Ausgabe-Bausteine, Flip-Flops, etc.)
- Definition zusätzlicher Bauteile durch Makrotechnik
- Makros können in Libraries gespeichert werden
- Interaktive Simulation mit Darstellung der Leitungszustände (d. h. Betätigung von 'Schaltern' mit der Maus und sofortige Reaktion der Schaltung)
- Erzeugen von Impulsdiagrammen
- Hardcopy-Funktion
- Hardwarevoraussetzung: ST/Monochrom-Monitor
- gängige Druckertreiber vorhanden

DM 89,–

NEU

ST ANALOG
SIMULATION VON ANALOGSCHALTUNGEN

Das Simulations-Programm zum Analysieren, Testen und Entwickeln von analogen Elektronikschaltungen (Kettenschaltungen) für Hobby, Ausbildung und Studium.
- Komfortable Maus-Steuerung
- Grafischer Schaltungsaufbau
- Einfache Eingabe und Änderung von Bauteilwerten
- Max. 65 Bauteile pro Schaltung (z. B. Stromquellen, Übertrager, Schwingkreise, offene und kurzgeschlossene Stichleitungen, Übertragungsleitungen usw.)
- „Wobbel-Generator" von 1 Hz bis 2 GHz
- Max. 0,1 Hz bzw. 1 Hz Auflösung
- Wahlmöglichkeit zwischen sehr schneller oder sehr genauer Berechnung
- Grafische Ausgabe von Spannungs- und Stromverhältnissen, von Phasenverläufen und von Eingangsimpedanzen
- logarithmische und lineare Koordinatenachsen
- Verstärkung und Offset einstellbar
- Einfaches Testen der Schaltung im Rückwärtsbetrieb
- Digitale Anzeige von Funktionswerten bei diskreten Frequenzen
- Hardcopy-Funktion
- Hardwarevoraussetzung: ST mit Monochrom-Monitor
- ausführliches deutsches Handbuch

DM 98,–

ST-MATH
DAS PROGRAMM FÜR SYMBOLISCHE ALGEBRA UND ANALYSIS

Das Programm ST-MATH ist ein Mathematikprogramm für den ATARI ST-Computer, das es Ihnen möglich macht, symbolische Mathematik auf Ihrem Computer zu betreiben, eine für Microcomputer seltene, für den ST ***einmalige*** *Anwendung.*
Mit den herausragenden Fähigkeiten des Programmes ST-MATH kann ein Schüler von Beginn der 8. Klasse an bis weit über das Abitur in die ersten Studiensemester vorteilhaft arbeiten. Ja, auch ein Achtkläßler kann dieses Programm bereits sinnvoll einsetzen, da die Kenntnis der höheren Mathematikfunktionen nicht Voraussetzung für das Arbeiten mit ST-MATH ist.
Also: Eine langfristige und wertvolle Anschaffung zu einem günstigen Preis.

- rechnet ohne Rundungsfehler
- verarbeitet symbolische Ausdrücke wie $2x + 3y = 5z$
- löst Gleichungen nach beliebigen Variablen auf
- beherrscht Grenzwerte, Differential- und Integralrechnung
- ideal für Matritzenrechnung
- unglaublich schnell, da **vollständig** in Assembler geschrieben
- nicht nur ein Mathematik-Programm, sondern gleichzeitig eine komplette KI-Sprache, die leicht erlernbar ist
- die Kombination von Mathematik-System und Programmiersprache ermöglicht auch Anfängern komplexe Mathe-Programme mühelos zu schreiben
- einfache und komfortable Bedienung
- für Schüler, die sich Fehlrechnungen ersparen wollen
- für Lehrer, die mit ST-MATH Klausuren mühelos und schnell korrigieren wollen
- für Studenten, die lange Umformungen und Rechnungen zeitsparend durchführen wollen
- für Ingenieure und sonst. Anwender, die oft komplexe nichtnumerische Probleme lösen müssen
- für Jeden, der eine preiswerte, aber dennoch vollwertige Sprache für künstliche Intelligenz sucht, die leicht zu erlernen ist
- Hardware: ATARI ST mit 512 KB und ROM-TOS oder 1 MB (dann auch RAM-TOS möglich), eins. Disk.-Laufw.
- arbeitet mit Farb- und SW-Monitoren
- Lieferung mit umfangreichem deutschen Handbuch, das die Mathematikfunktion detailliert erläutert und zugleich eine ausführliche Einführung in die Programmierung von ST-MATH enthält.
- Ausführliches Informationsblatt über den Umfang und die Möglichkeiten von ST-MATH kostenfrei beim Verlag erhältlich.

Ein Spitzenprogramm **DM 98,–**

Achtung: Version 2.1 – Austausch

Alle ST-MATH-Besitzer können die aktuelle Version 2.1 ab sofort erhalten. Senden Sie **nur** die Originaldiskette zurück (kein Handbuch) und legen Sie für den Versand 5,– DM in Briefmarken bei.

* alle Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ______________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

Auslieferung in der Schweiz:
DataTrade AG
Langstr. 31
CH-8021 Zürich

Blautönen. Erstellen Sie verschiedene "Prototypen" von Sternen und verlassen Sie die Lupe dann mit zweimaligem Klikken der rechten Maustaste.

Im Auswahlmenü von DENISE finden Sie das Icon *"Quellbild"*. Klicken Sie auf dieses Icon und wählen Sie einen Ihrer Sterne-"Prototypen" mit einem möglichst kleinen Rechteck aus. Mit dem Icon *"Kopieren"* im Zeichenmenü können Sie diesen Stern beliebig oft mit einem Mausklick links in Ihr Bild kopieren. Wie immer beenden Sie die Funktion mit einem Druck auf die rechte Maustaste.

Jetzt streuen Sie noch nach Belieben mit der Sprühdose einzelne Punkte in das Bild: Stellen Sie dazu die Zeichenfarbe ein: Klicken Sie auf das Icon *"Farbpalette"*. Jetzt müssen wir allerdings unterscheiden, ob Sie an einem Monochrom- oder an einem Farbmonitor arbeiten:

Im Monochrombetrieb sehen Sie jetzt 16 Felder für die "Zeichenfarbe", neben reinem SCHWARZ und WEISS noch 14 weitere Felder, die spezielle Graumuster nach dem *"Ordered Dither"*-Verfahren enthalten, um damit in der hohen Auflösung Schattierungen darzustellen. Klikken Sie in das Feld für die Farbe "WEISS".

Im Farbbetrieb klicken Sie auf ein hellblaues Feld der Farbpalette. Es wird daraufhin weiß umrandet. Verlassen Sie die Farbauswahl über den Button *"EXIT"*.

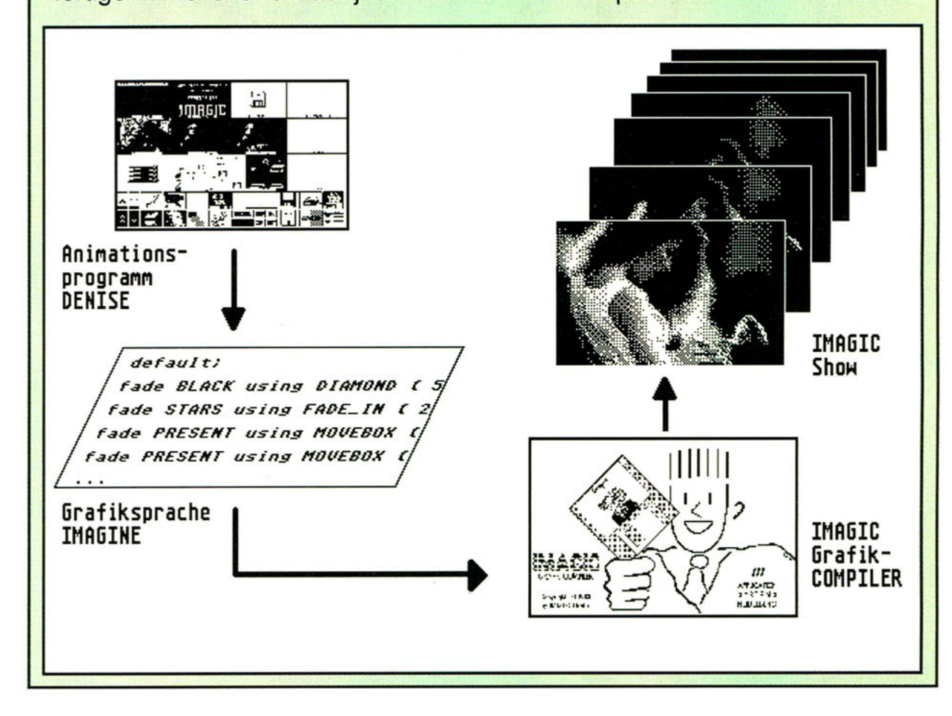
Stellen Sie den Zeichenmodus auf *"Normal"* (Klick auf "NORMAL") und setzen Sie dann die Sprühdose ein. Ziehen Sie jetzt mit der Maus den Sprühradius, dargestellt als Kreis, zur vollen Bildschirmgröße auf und klicken Sie einmal kurz die linke Maustaste. Wenn Sie jetzt kurz und vorsichtig mit der Maus klicken, streuen Sie kleine Punkte überall ins Bild. Wollen Sie eine Milchstraße darstellen, halten Sie die linke Shifttaste fest und bewegen die Maus beim Sprühen entlang einer leicht gekrümmten Bahn. Verlassen Sie die Sprühdose mit Druck auf die rechte Maustaste.

Ist Ihr Sternenhimmel fertig erstellt, drükken Sie die Taste <RETURN>, um das Bild automatisch in die Bilddatenbank von DENISE zu speichern. Klicken Sie in das Feld *"NEU"* der Dialogbox. Mit dem Icon *"Ausgang"* rechts unten im Menü des Zeichenteils gelangen Sie zurück in die Bilddatenbank.

Soviel zur Erstellung unseres ersten Bildes, dem Sternenhimmel. Auf der *Diskette 2* finden Sie einen bereits fertig erstellten Sternenhimmel als Bild *"STARS.IC1"*. Laden Sie dieses Bild von der Diskette 2, indem Sie einfach auf ein freies Feld der DENISE-Bilddatenbank klicken und in die Fileselektorbox den Namen "STARS.IC1" eintragen.

Das IMAGIC-Konzept

Bei der Arbeit mit IMAGIC ist immer wieder vom "Grafikcompiler" die Rede. Hier arbeiten wir jedoch die ganze Zeit mit einem Grafikprogramm unter Mauskontrolle, ohne die Spur einer Computersprache zu erkennen. Zur Erläuterung: Der Filmeditor von DENISE ist ein übergeordnetes Werkzeug, das mit Hilfe der Maus die Befehle der Grafiksprache IMAGINE auf einem Storyboard arrangiert, also ein "grafischer Interpreter" für IMAGINE. Die Sprache ist jedoch weitaus vielseitiger, als es sich im Filmeditor von DENISE offenbart. Viele Befehle lassen sich daher nicht mehr auf einfache Weise grafisch eingeben. Deshalb kann DENISE die Anweisungen aus dem Filmeditor direkt in der Grafiksprache IMAGINE auf Diskette speichern, so daß Sie die Möglichkeit haben, mit jedem beliebigen Texteditor Ihr "IMAGINE-Programm" zu überarbeiten und die Animation bis ins kleinste Detail auszufeilen. Der COMPILER aus dem IMAGIC-Paket (er ist bei der PD-Version nicht enthalten) "übersetzt" Ihre Anweisungen sodann, bindet die notwendigen Einzelbilder dazu und erstellt so eine komplette IMAGIC-Show. Ein frei kopierbares Run-Time-Modul von nur 32 KBytes Länge spielt dann die fertige IMAGIC-Show auf jedem ATARI ST Computer ab.



Sterntaler

Jetzt setzen wir das Ergebnis unserer Arbeit als Einstieg für den Computerfilm ein. Wir schreiben die ersten Anweisungen in unserem elektronischen Storyboard von DENISE. Im Detail machen wir folgendes: Wir beginnen den Film mit einer einfachen, vom Computer errechneten Überblendung nach Schwarz. Damit beginnt der Film immer mit einem schwarzen Bild, auch wenn er später in einem Endloskreis läuft. Dann werden zwei parallel laufende Farbanimationen aktiviert, die unsere Sterne in unterschiedlichem Rhythmus blinken lassen. Das wirkt deutlich natürlicher als eine einzige Farbanimation. Jetzt wird der gesamte Sternenhimmel insgesamt aufgeblendet, die einzelnen Phasen der Aufblendung werden dabei ebenfalls vom Computer errechnet.

Ans Werk: Wenn Sie jetzt Ihr selbst erstelltes Sternenbild anstelle von "STARS.IC1" einsetzen möchten, sollten Sie, wenn Sie in Farbe arbeiten, noch die Farbpaletten der beiden Bilder einander anpassen, damit die folgenden Anweisungen zur Farbanimation übereinstimmen. Laden Sie die Datei "STARS.IC1", klikken Sie danach einmal auf das verkleinerte Bild von "STARS". Sie haben das Bild korrekt ausgewählt, wenn das Minibild

invertiert ist. Klicken Sie jetzt auf das Icon "Farbpalette", dann in der Farbauswahl auf den Button *"SPEICHERN"*. Verlassen Sie die Farbauswahl über *"EXIT"* und wählen Sie jetzt Ihr eigenes Sternenbild aus. Gehen Sie wieder in die Farbauswahl und klicken Sie diesmal den Button *"LADEN UND ANPASSEN"*. DENISE rechnet daraufhin die Farbvektoren in Ihrem eigenen Bild automatisch um, so daß es in der ursprünglichen Farbpalette von "STARS" dargestellt wird. Bestätigen Sie das Ergebnis mit Druck auf die linke Maustaste und verlassen Sie die Farbauswahl.

Mit der *Diskette 1* in Laufwerk A klicken Sie auf das Icon *"Filmen"*, der Filmeditor von DENISE wird initialisiert. Nach kurzer Zeit sehen Sie vier Menüzeilen. Die oberste Zeile, das *"Clipboard"*, dient zum Ablegen und Zwischenspeichern einzelner Anweisungen. Die zweite Menüzeile ist jetzt noch vollständig leer: das elektronische "Storyboard". Diese Zeile gleicht einem Ausschnitt aus einem langen Filmstreifen, mit den Scrollpfeilen "<" und ">" kann man den Ausschnitt auf dem Streifen verschieben, wenn der Film länger wird. Die beiden unteren Menüzeilen enthalten die Kommandos, die zur Bedienung des Filmeditors notwendig sind. Klicken Sie auf das Icon *"Zeit"*. Das Symbol "hängt" jetzt an der Maus, und Sie können es über den Bildschirm bewegen. Schieben Sie das Icon auf die erste freie Stelle im Storyboard und legen es dort mit einem Linksklick ab. Es erscheint eine Box: *"PAUSE: 0___"*. Bestätigen Sie die Eingabe "0" mit Druck auf die Taste <RETURN>. Damit haben Sie die Filmgeschwindigkeit auf Maximum eingestellt, denn der Anfang unseres Films soll möglichst ohne Pause ablaufen.

Wählen Sie ein schwarzes Bild aus, indem Sie auf das Icon *"Farbpalette"* und darin auf das Feld *"SCHWARZ"* klicken. Sofort danach klicken Sie auf das Icon *"Effekt"*. Es läßt sich jetzt verschieben, Sie legen es an der Position 2 des Filmstreifens ab. DENISE zeigt Ihnen daraufhin eine Auswahl aus seinem Sortiment von Überblendeffekten: Wählen Sie den Effekt *"DIAMOND"* aus der untersten Zeile mit einem Mausklick aus. Jetzt erscheint eine Dialogbox, in der Sie die Steuerparameter für den Effekt einstellen können. DENISE schlägt Ihnen hier bereits die Parameter in Standardform vor. Lassen Sie die Parameter fürs erste einmal unverändert und bestätigen Sie die Auswahl mit Druck auf die Taste <RETURN>. Sie sehen wieder den Filmeditor, doch das Icon *"Effekt"* wurde durch das Icon *"Diamond"* ersetzt. Diese Eingabe bedeutet für DENISE:

"fade BLACK using DIAMOND (50, 1)"

- Sie haben die erste Anweisung der Grafiksprache IMAGINE mit der Maus eingegeben. Später zeigen wir Ihnen, wie Sie alle Anweisungen aus dem Storyboard sogar automatisch in lesbarer Textform abspeichern können. Mit dem IMAGIC Grafikcompiler, den es im Originalpaket von IMAGIC gibt, werden diese Anweisungen dann in die Steuerbefehle für eine eigenständig laufende IMAGIC-Show übersetzt.

Wenn Sie einen Befehl aus dem Filmeditor löschen wollen, klicken Sie auf das entsprechende Symbol im Filmstreifen und verschieben das Symbol dann über *"OSCAR"*, den Mülleimer. Mit einem Linksklick wird das Symbol gelöscht.

Die Anweisungen aus dem folgenden Abschnitt benötigen Sie nur, wenn Sie im Farbmodus des ATARI ST arbeiten, denn jetzt legen wir die Farbanimation fest. Als erstes wird die Farbpalette unseres Sternenhimmels gesetzt, ohne dabei das Bild selbst zu zeigen: Wählen Sie das Bild "STARS" oder Ihr selbst erstelltes Bild aus, indem Sie es wie bekannt mit der Maus anklicken. Sollten Sie Ihr Bild gerade nicht als Minibild sehen können, können Sie mit den vertikalen Scrollpfeilen ganz links in der Bilddatenbank hin- und herscrollen. Klicken Sie wieder auf das Icon *"Effekt"* und schieben es an die dritte freie Stelle im Storyboard. Wählen Sie den Effekt *"SUNRISE"* aus der obersten Zeile rechts aus. Jetzt folgt ein kleiner Kniff: Damit *"SUNRISE"* nur die Farbpalette setzt, ohne jedoch etwas auf den Bildschirm zu zeichnen, klicken Sie mit der Maus in die Parameterzeile *"ANZAHL"* und setzen den Wert dort auf "0". Bestätigen Sie Ihre Eingaben mit <RETURN>. Dieser Kniff ist notwendig, weil bei eingeschalteter Farbanimation IMAGIC Farbpaletten einzelner Bilder nicht mehr setzt, um die Farbanimation nicht zu stören.

Legen Sie nun zwei Farbanimationen fest: Klicken Sie auf das Icon *"Modul"* und schieben es an die nächste freie Position im Storyboard. Wählen Sie das Modul *"COLOR_ROT"* aus. Als Parameter geben Sie ein:

Slot = 0, Pause = 20, F05 = 6, F06 = 7,
F07 = 5.

Alle anderen Zeilen bleiben auf "-1". Damit werden die Farbregister 5, 6, 7 jeweils alle 1/5 (20/100) Sekunden vertauscht. <RETURN> bestätigt die Eingabe. Schieben Sie ein weiteres *"COLOR_ROT"* in das Storyboard. Die zweite Farbrotation wird mit

Slot = 1, Pause = 30, F13 = 14, F14 = 13

eingegeben. Mit den seitlichen Scrollpfeilen können Sie bei der Parametereingabe die nicht sichtbaren Zeilen erreichen.

Ab hier folgen wieder Anweisungen, die Sie auch im Monochrombetrieb ausführen können: Klicken Sie noch einmal auf das Minibild Ihres Sternenhimmels. Schieben Sie das Icon *"Effekt"* hinter die letzte belegte Stelle im Storyboard, wählen Sie den Effekt *"FADE_IN"* aus der zweiten Zeile von unten aus. Bestätigen Sie die Parameter einfach mit Druck auf die Taste <RETURN>.

Soviel zu den ersten Anweisungen an den

Clipboard

Storyboard

Parameter zurücksetzen · Pause Null · Schwarz beginnen · Sterne einblenden · Schriften in das Bild schieben

Bedienungselemente

Bild 5: Elektronisches Storyboard

Filmeditor von DENISE. Auch wenn die Eingaben am Anfang noch etwas ungewohnt sind - nach kurzer Zeit werden Sie feststellen, wie einfach und komfortabel Sie mit dieser Technik die Befehle für Ihren Computerfilm erstellen können.

Doch jetzt heißt es “Film ab!”: Klicken Sie auf das Icon *“Start”*. Die erste kleine Sequenz aus unserem Computerfilm erscheint!

Als die Buchstaben laufen lernten

Alles ok? Dann geht’s gleich weiter mit dem nächsten Teil: Ein Schriftzug in überdimensional großen Buchstaben schiebt sich ins Bild. Laden Sie bitte das Bild *“PRESENT.IC1”* in die DENISE-Bilddatenbank durch einen Klick auf ein freies Feld. Klicken Sie sodann zweimal links auf das Minibild. Es “hängt” an der Maus. Schieben Sie es in das Feld ganz oben rechts der Bilddatenbank (Arbeitsbuffer 1) und legen es dort mit einem Linksklick ab. Dadurch können wir das Bild im Filmeditor zur Vermessung verwenden.

Klicken Sie noch einmal auf das Minibild “PRESENT” aus der Bilddatenbank (nicht das Bild im Buffer 1!) und dann auf das Icon *“Effekt”*. Schieben Sie das Effekt-Icon an die nächste freie Stelle im Storyboard. Bevor das Storyboard den rechten Bildschirmrand erreicht hat, können Sie es mit den horizontalen Scroll-Pfeilen verschieben. Jetzt wählen Sie den Effekt *“MOVEBOX”* aus, der ein ausgewähltes Rechteck über den Bildschirm verschiebt. In die erste Zeile der Parameterbox geben Sie *“M_TRANSP”* ein. Klicken Sie dann mit der linken Maustaste in die zweite Zeile der Box (Parameter “X1”). Die Zeile wird daraufhin invertiert. Klicken Sie anschließend auf den Button *“Fenster”*. Jetzt sollte das Bild “PRESENT” in voller Größe angezeigt werden, zusammen mit einem Fadenkreuz, das Sie mit der Maus positionieren können. Legen Sie jeweils durch Linksklick die linke obere und die rechte untere Ecke eines Rahmens um die gesamte erste Zeile “APPLICATION SYSTEMS HEIDELBERG” fest. Sie kehren in die Parameterbox zurück, dabei werden die Koordinaten in die folgenden vier Parameterzeilen übernommen.

Jetzt geben Sie noch Start- und Endpunkte der Bewegung sowie die Anzahl der Schritte ein. Die Parameterzeilen sollten in etwa so aussehen:

```
Modus = M_TRANSP, X1 = 68, Y1 = 64,
X2 = 940, Y2 = 314, START_X = 68,
START_Y = 350, END_X = 68
END_Y = 64, SCHRITTE = 360
```

Damit bewegt sich die erste Zeile langsam von oben ins Bild. Bestätigen Sie die Parameterauswahl mit der Taste <RETURN>.

Für die Bewegung der nächsten beiden Zeilen benötigen wir zwei weitere Aufrufe von *“MOVEBOX”* mit folgenden Parametern:

```
Modus = M_TRANSP, X1 = 237,
Y1 = 369, X2 = 765, Y2 = 529,
START_X = 237, START_Y = 1000,
END_X = 237, END_Y = 369,
SCHRITTE = 60
```

sowie

```
Modus = M_TRANSP, X1 = 80, Y1 = 599,
X2 = 912, Y2 = 954, START_X = 80,
START_Y = 1000, END_X = 80,
END_Y = 599, SCHRITTE = 120
```

Da IMAGIC den Bildschirm als Matrix von 1000 * 1000 Punkten betrachtet, sind die Parameter unabhängig von der Auflösung des ATARI ST; IMAGIC-Animationen können somit leicht von einer in die andere Auflösung umgesetzt werden.

Starten Sie jetzt wieder den Film. Das Intro sollte wie erwartet ablaufen.

Dieses war der erste Streich,...

Jetzt folgt der nächste Teil unseres Computerfilms: Das erste Bild wird abgeblendet, die Farbanimation ausgeschaltet. Ein Frauengesicht mit einer Sonnenbrille erscheint. Sodann wird nur der Hintergrund ausgeblendet, zwei Linien erscheinen aus den Gläsern der Sonnenbrille, der ganze Kopf wird nach rechts geschoben. Nach einer kurzen Verzögerung erscheinen weitere Linien im Bild, dadurch entsteht der Eindruck einer weiten, im Raum liegenden Ebene.

Ans Werk: Klick auf Arbeitsbuffer 1, Effekt-Icon ins Storyboard, Auswahl Effekt *“FADE_OUT”*, Parameter: *Pause = 200, Endwert = 0*. Danach Modul *“COLOR_STOP”* anfügen.

Mit dem *“Ausgang”*-Icon verlassen Sie den Filmeditor, jetzt laden Sie in der DENISE-Bilddatenbank das Bild *“GLASSES.IC1”*. Das Bild, das so unverändert von der Digitalisierung kommt, muß noch bearbeitet werden. Das geladene Bild wird in den Arbeitsbuffer 1 kopiert. Im Zeichenteil von DENISE wählen Sie den *“RADIERER”*. Mit der Maus läßt sich jetzt ein beliebig großer “Radiergummi” bestimmen. Radieren Sie den Hintergrund möglichst gründlich aus dem Bild, halten Sie dabei die linke Maustaste gedrückt. Für die Feinarbeit wählen Sie unterschiedlich große Formen von Radiergummis aus. Nach Abschluß der Arbeiten wird das Bild mit Druck auf <RETURN> und Klick auf *“NEU”* in die Datenbank eingespeichert. Das fertige Bild können Sie auch als *“GLASS_BL.IC1”* von Diskette laden.

Im Filmeditor ergänzen Sie jetzt den Film mit folgenden Anweisungen: Bild *“GLASSES”*, Effekt *“FADE_IN”*, Parameter: *Pause = 700, Startwert = 0*.

Danach Bild *“GLASS_BL”*, Effekt *“BITS”*. Das Bild wird aufgeblendet, der Hintergrund verschwindet.







Jetzt ziehen Sie zwei blaue Linien, indem Sie den IMAGIC-Effekt *"PLOTLINE"* einsetzen. Kopieren Sie das Bild *"GLASSES"* in den Arbeitsbuffer 1. Im Bild *"GLASSES"* ist die Farbe Blau dem Farbregister 10 zugeordnet. Klicken Sie auf das Icon *"Farbpalette"* und dann auf die Farbe *"Hellrot"*, die dem Register 10 entspricht. Arbeiten Sie in Schwarzweiß, wählen Sie die Farbe *"Weiß"*. Sofort anschließend schieben Sie wieder das Effekt-Icon in das Storyboard. Wählen Sie also *"PLOTLINE"*, Parameter: *ZM = 0, LM = FULL*. Anfangspunkt (X1, Y1) und Endpunkt (X2, Y2) der Linie ermitteln Sie, indem Sie die Parameterzeile und danach den Button *"PUNKT"* anklicken. Die Werte sollten ungefähr lauten: *X1 = 190, Y1 = 344, X2 = 0, Y2 = 344.*

Die zweite Linie entsteht, indem Sie einfach das Effekt-Icon der ersten Linie auf dem Storyboard duplizieren: Klicken Sie es an und legen Sie es gleich wieder mit einem Linksklick auf dem Storyboard ab. Jetzt machen Sie einen Doppelklick auf das so kopierte Effekt-Symbol. Sie sehen jetzt die bereits ausgefüllte Parameterbox. Ändern Sie die Parameter in *ZM = 0, LM = FULL, X1 = 259, Y1 = 364, X2 = 0, Y2 = 364.*

Starten Sie den Film durch Klick auf das Icon *"Start"*. Nachdem er abgelaufen ist, legt DENISE das letzte Bild des Films im Buffer 2 (SCREEN) ab. Kopieren Sie dieses Bild jetzt in eine freie Stelle der Bilddatenbank, denn Sie brauchen es noch!

Jetzt wird der ganze Kopf nach rechts aus dem Bild geschoben:

Klicken Sie auf das gerade in der Bilddatenbank gesicherte Bild, wählen Sie dann den Effekt *"SMEARBOX"*, Parameter:

Modus = 0, X1 = 80, Y1 = 53, X2 = 796,
Y2 = 999, START_X = 80, START_Y = 53
END_X = 1000, END_Y = 53
SCHRITTE = 40

Weil *"SMEARBOX"* den gewählten Bildausschnitt nicht versetzt, sondern "verschmiert", bleiben die blauen Linien wie gewünscht nach der Bewegung im Bild stehen. Jetzt folgen 5 weitere blaue Linien, die Symbole werden wieder durch einfaches Kopieren erzeugt. Mit einem anschließenden Doppelklick ändern Sie die Parameter so ab, daß eine Linie unter die nächste gezogen wird, mit immer größerem Abstand. Damit wird der Bildschirm zu einer Ebene im leeren Raum.

Jetzt heißt es wieder "Film ab"! Wir haben die erste Phase unseres Computerfilms erstellt. Mit ein wenig Routine benötigen Sie dafür nicht länger als 15 Minuten.

Sie können die erste Phase dieses Films auch komplett in DENISE einladen: Im DENISE-Desktop (zweimal auf *"Ausgang"*-Icon klicken) finden Sie unter dem Menüpunkt *"Grafik"* einen Eintrag *"STR laden"*, (Struktur laden). DENISE lädt mit einer Struktur automatisch alle benötigten Bilder und das komplette Storyboard einer Animation von Diskette. Mit der Originalversion von IMAGIC können Sie solche Strukturen auch abspeichern. Laden Sie *"C_INTRO.STR"* im Farbmodus bzw. *"M_INTRO.STR"* im Monochrommodus. Beachten Sie bitte: Während *"STR laden"* vorher alle Bilder der DENISE-Bilddatenbank aus dem Speicher löscht, können Sie mit *"STR zuladen"* eine Struktur nachladen und so miteinander verknüpfen. Der neue Filmteil wird im Filmeditor an den vorigen Teil angefügt, getrennt durch ein Symbol *"STOP"*.

Soviel also für diesmal. Im nächsten Teil unserer Serie erstellen wir weitere Teile des Films und zeigen Ihnen, wie die digitale Bildfilterung und weitere ausgewählte Teile des Programms funktionieren.

Bis dahin wünschen wir Ihnen ein frohes Fest und einen guten Rutsch ins neue Jahr

Alexander Beller & Jörg Drücker

Vorbereitung der IMAGIC PD-Disketten

1. Erstellen Sie eine Sicherheitskopie von beiden Disketten, wie wir es in der letzten Ausgabe bereits geschildert haben. Wichtig: Aus Platzgründen sind beide Disketten einseitig in einem Spezialformat von *10 Sektoren je Track* erstellt. Kopieren Sie die Disketten dateiweise, wenn Ihnen kein geeignetes Kopierprogramm zur Verfügung steht. (Das Spezialformat können Sie mit dem Programm DENISDEM.PRG selbst erstellen, im Menüpunkt *"Formatieren" aktivieren* Sie den Button *"extended"*).

Arbeiten Sie im folgenden immer nur mit der Sicherheitskopie !

2. (Nur für Anwender, die mit einem Monochrommonitor arbeiten):

Löschen Sie von der Diskette 2 (PD 171) alle Dateien, die mit *"C_"* beginnen und die Endung *".STR"* haben. Auf der Diskette 1 (PD 170) befindet sich ein Ordner *"MONO.STR"*: Öffnen Sie diesen Ordner und kopieren Sie alle Dateien aus dem Ordner auf die Diskette 2. Kopieren Sie die Dateien direkt in das unterste Directory von Diskette 2, ohne den Ordner selbst mitzukopieren!

Zur Erläuterung: Dateien mit der Endung *"*.STR"* enthalten alle Informationen für ein komplettes Storyboard von DENISE in Kurzform. Aus Platzgründen hatten jedoch nicht mehr alle Strukturdateien für Farbe und Schwarzweiß auf der Bilderdiskette Platz. Deshalb müssen Sie die Strukturdateien für Farbe durch die entsprechenden Dateien für Schwarzweiß ersetzen.

Ein Hinweis für diejenigen von Ihnen, die ein Original von IMAGIC besitzen: In der Serie BILDWERKSTATT ATARI ST entsteht ein Computerfilm, den Sie natürlich mit Ihrer Originalversion ebenfalls bearbeiten können. In dem Film kommen zwei neue IMAGIC-Effekte (DISC und DIAMOND) zum Einsatz, die erst ab der Version 1.1 von IMAGIC zur Verfügung stehen. Daher können Sie mit IMAGIC 1.0 den Computerfilm dieser Serie nicht abspielen. Sollten Sie also noch eine Version 1.0 von IMAGIC besitzen, setzen Sie sich bitte mit APPLICATION SYSTEMS HEIDELBERG wegen eines Updates zu 1.1 in Verbindung. Zusammen mit dem Update erhalten Sie auch die digitalisierten Bilder dieser Serie.

Das Bildmaterial dieses Computerfilms ist durch Digitalisierung entstanden (s. ST-Computer 12'88). Die Bilder wurden teilweise mit dem Zeichenprogramm DENISE aus dem IMAGIC-Grafikpaket überarbeitet.

Die übergroßen Schriften stammen aus dem Programm ART DIRECTOR, sie wurden mit den Blockoperationen von DENISE passend vergrößert, zusammengesetzt und farblich angepaßt.

Alle Animationen und Filmsequenzen wurden mit dem Zeichenteil und dem Filmeditor von DENISE erstellt.

PHÖNIX^ST

Der Durchbruch in Sachen Textverarbeitung

Wenn Sie einen 'normalen' Brief erstellen möchten, soll das genauso wenig ein Problem sein, wie das Dokument unten. Diese Zusatzfunktionen sollen aber die Bedienung nicht komplizierter machen. Selbst wenn Sie aufwendige Dokumente erstellen, müssen Sie die Funktionen weder im Kopf haben, noch in einem 1000 seitigen Handbuch danach suchen.

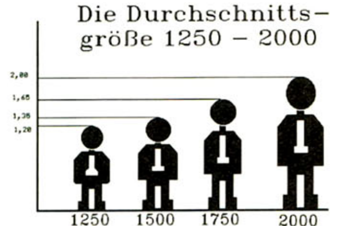
Phönix** ist so einfach und verständlich in der Bedienung, daß Sie in der Regel bei der zweiten Anwendung schon kein Handbuch mehr brauchen. Auf selbstverständliche Textverarbeitungsfunktionen wie z.B. Suchen, Ersetzen, Blockoperationen etc. wollen wir an dieser Stelle nicht eingehen, diese Funktionen sind einfach vorhanden. **Die erstellten Texte sehen Sie auf dem Bildschirm genauso wie später beim Druck. (WYSIWYG)

Phönix^ST DM 198.-





Wo wachsen wir hin?



Universität Kamerun, Dekan Umba-Wumba - ein Auszug aus seiner neuen Disertation:

Das Wachstum des Menschen hat in den letzten 1000 Jahren rapide zugenommen. Bessere Lebensbedingungen, Fortschritte der Medizin und mehr Luxus haben einen immensen Größenzuwachs begünstigt. Hatte der Mensch des 11. Jahrhunderts noch Probleme ohne Hilfe auf ein Pony zu steigen, muß der Mensch des 20. Jahrhunderts schon aufpassen, nicht darüber zu stolpern. Mit der Körpergröße wuchs auch die Gehirnkapazität. Man erfand das Schiesspulver, das Radio, Einstein entwickelte seine Relativitätstheorie und McDonalds gründete sein Hamburgerimperium. Überall hielt die technische Revolution Einzug, in der Unterhaltung, der Freizeit und bei den Fortbewegungsmitteln. Nur ein Bereich wurde sehr lange vernachlässigt: Die zwischenmenschliche Kommunikation. Auch heute werden Briefe noch lustlos mit der Hand gekritzelt und längere Texte mit Gewalt in die Schreibmaschine gehämmert. Illustrationen, Schriftwechsel und allgemeines Layout werden vermieden oder irgendwie improvisiert. Von Fortschritt, trotz der Revolutionen nicht zuletzt auch im Computersektor, also keine Spur. Das muß nicht sein! Mit Phönix wurde ein Programm geschaffen, mit dem es Spaß macht, Briefe zu gestalten und Dokumente zu erstellen. Die Bedienung ist einfacher als die mancher Schreibmaschine und die Ergebnisse sind einfach überwältigend. Wir wissen nicht, was Ihnen Ihr freundlicher Nachbar empfiehlt, wir empfehlen Phönix!

Erster Phönix auf dem Mond eingesetzt

Allgemeine Einigkeit der Mondnationen: Phönix ist Spitze

Als erstes Textsystem auf dem dritten Mond der Venus konnte sich Phönix durchsetzen. Durch seine einfache Bedienbarkeit und die Möglichkeit unterschiedlichste Fonts zu laden, kann es von allen auf dem Mond ansässigen Völkern gleichsam genutzt werden. Ob nun die chinastämmigen Ostmonder Kanji von oben nach unten schreiben wollen, die arabischstämmigen Mittelmonder Sanskrit von rechts nach links oder die eurasischen Südmonder lateinisch oder russisch von links nach rechts, alles geht. Was sich nicht in Wort und Schrift ausdrücken lässt, dafür haben sich unsere Mondbewohner ein Symbolsystem ausgedacht. Dieses umfasst bereits Symbole für Steak, Whisky (Bourbon und Scotch getrennt), Wodka und Eierpfannkuchen.



Bier und Whiskey auf Südmondisch

Die Leistungen von Phönix:

ASCII Texte (von anderen Textprogrammen) einlesen, einfügen und ausgeben - Anzeigemöglichkeiten auf dem Bildschirm 50%, 75%, 100%, 150%, 200% der Originalgrösse - 9 Zeichensätze werden mitgeliefert, die in vielen Grössen benutzbar sind, zusätzlich kann jeder Zeichensatz in jeder Grösse fett, kursiv, unterstrichen, invertiert, hoch- und tief gestellt werden. Ihren Text können Sie auch absatzweise linksbündig, rechtsbündig, im Blocksatz oder zentriert formatieren. Für die Erstellung von Dokumenten mit mehreren Spalten kann die Anzahl und Grösse der Spalten beliebig sein. Die Spaltengrösse kann mit der Maus oder durch Eingabe in mm definiert werden. Die Fliessrichtung des Textes über mehrere Spalten können Sie frei definieren. Wenn Sie die Grösse einer Spalte nachträglich verändern, wird der Text in diesem Abschnitt automatisch umformatiert. Phönix trennt nach deutschen Trennregeln. Bilder können Sie überall im Text plazieren. Jeder 9 oder 24 Nadel Matrixdrucker kann verwendet werden. Wenn Sie lieber mit einem Laserdrucker arbeiten möchten, können Sie einen HP Laserjet + kompatiblen benutzen. Phönix ist ein deutsches Programm.

Ein ausführliches Handbuch mit allgemeiner Einführung in Desktop Publishing ist genauso selbstverständlich wie unser Update Service und telefonische Beratung.

COUPON

Senden Sie mir bitte:
☐ Phönix für den Atari ST

Diskettenformat ☐ einseitig ☐ doppelseitig
Drucker ☐ 9 Nadel ☐ 24 Nadel ☐ Laser

zzgl. DM 6.- Versandkosten (unabhängig von der bestellten Anzahl) per
☐ Nachnahme ☐ Scheck liegt bei

Meine Adresse: ______________________

____

____

RAAB Datentechnik
Friedhofstr. 36 - 8605 Hallstadt
Telefon 0951-73061 - Telefax 0951-73068

RELAX

RELAX sucht das Spiel des Jahres 1988

Herzlich willkommen zur ersten Ausgabe von RELAX im neuen Jahr. Das alte Jahr ist vorbei, und doch hat es viele Erinnerungen hinterlassen. Ich meine damit die vielen Topspiele, die uns die Softwarehäuser 1988 beschert haben. Erinnern Sie sich noch an die vielen ausgezeichneten Actionspiele, Adventures, Rollenspiele und Simulationen, die Sie im letzten Jahr fasziniert haben? Hervorragend, wir suchen nämlich das Spiel des Jahres 1988. Bitte schreiben Sie Ihre beiden Lieblingstitel auf eine Postkarte und schicken Sie sie an:

MAXON Computer GmbH
ST-Computer
Kennwort: Spiel des Jahres
Industriestr. 26
6236 Eschborn

Einsendeschluß: 31.01.89

Unter allen Einsendern verlosen wir Spieleüberraschungspakete von Bomico. Wie immer sind Mitarbeiter des Heim Verlags und der MAXON Computer GmbH von der Verlosung ausgeschlossen.

Viel Spaß und Glück!

NEWS

"TRIAD" heißt eine neue Compilation von Mirrorsoft für ATARI ST. Drei ehemalige Hitparadenstürmer, nämlich Starglider, Defender of the Crown und Barbarian (Psygnosis), gibt es für knapp 100 DM.

Von Coktel Vision gibt es ein Spiel für 4-8jährige Computerspieler. In" **PETER PAN"** muß der junge Spieler Gegenstände mit der Maus auf dem Bildschirm suchen und Figuren durch Labyrinthe bewegen. Eine spaßige Angelegenheit,



zumal das Game über eine exzellente Sprachausgabe verfügt. Für Spieler älteren Semesters ist das Spiel allerdings nicht geeignet, da es viel zu leicht ist.

"FERNANDEZ" ist ein Game für hartgesottene Helden à la Stallone. Fernandez, ein räudiger Diktator, hat sich in seinem Hauptquartier verbarrikadiert und läßt seine Soldaten in einem riesigen Gebiet Patrouille schieben. Der Diktator muß sterben, damit wieder Ruhe und Ordnung in den Staat einkehren kann. Deshalb wird der Spieler beauftragt, mit einem Jeep ins Feindesland zu fahren und auf alles zu schießen, was sich bewegt. Während der Kampfhandlung müssen Kriegsgefangene befreit und Munitionslager gestürmt werden.

"PEPSI MAD MAX CHALLENGE GAME" von U.S. Gold versetzt Sie in die Urzeiten der Computer- und Videospielgeschichte zurück: PAC MAN!
Simple Grafik, nervtötende Soundeffekte und eine fehlende Joysticksteuerung sind ihr Geld nicht wert.

Ebenfalls von Coktel Vision stammt **"FREEDOM"**. Sie sind Anführer rebellierender Sklaven und müssen versuchen, andere Sklaven zum Aufstand zu bewegen. Dabei hetzen Sie bissige Doggen und



Aufseher mit Peitschen und Säbeln. Wie bei den Spielen von Coktel gewohnt, gibt es hervorragende Grafiken zu bewundern. Trotzdem konnte mich Coktels neues Adventure nicht überzeugen.

"LOMBARD RALLYE" heißt eine exzellente Rallyesimulation aus dem Hause Mandarin Software. Rasante 3D-Grafik, eine realistische Cockpitgrafik und die Möglichkeit, beschädigte Autos zu reparieren, bereiten eine Menge Spaß.

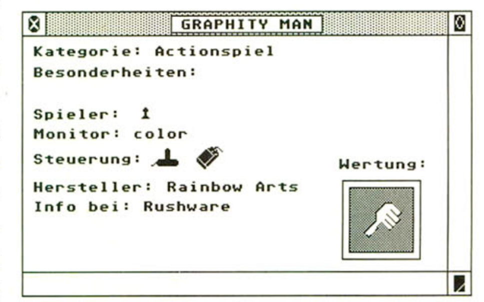


Graphity Man
Wilde Sprühorgie!

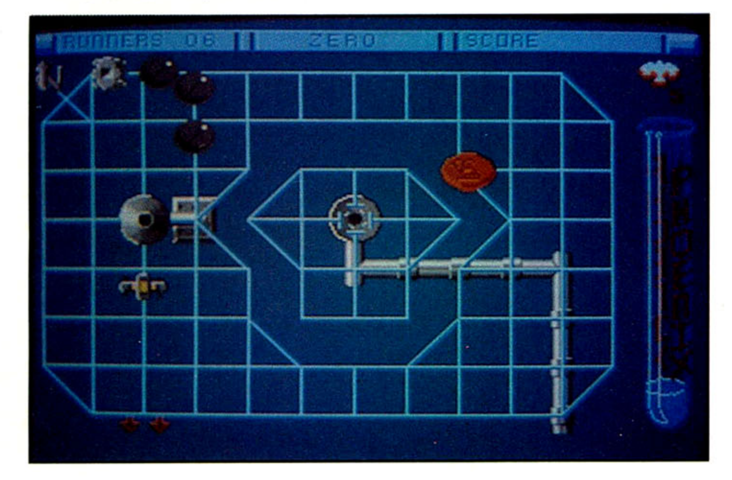


"Rainbow Arts" stand auf dem Ansteckbutton, der mir beim Öffnen der Verpackung von Graphity Man entgegenplumpste. Ich spießte die Nadel an meine Brust und stopfte die Diskette ins Laufwerk. Zur Begrüßung spielt der ST eine spritzige Titelmelodie, und dann geht es auch schon in die Welt der Spraydosenkünstler. Der Spieler übernimmt die Rolle von Mickey, der in einem öden Viertel voller Betonbaracken lebt. Überall, wo Mikkey hinsieht, erblickt er grauen langweiligen Beton (Gähn!) "Den müßte man ein bißchen veredeln", denkt Mickey, schnappt sich Spraydosen inklusive Farbe und begibt sich auf die gefährliche Straße. Dort lauern viele Gefahren für Burschen mit Spraydosen in der Hand. Spießbürger und vor allem die Polizei hassen Jungs, die die Wände mit irgendwelchem Schmierkram vollsprühen. Da werfen Omas mit Blumentöpfen, Opas schlagen mit dem Spazierstock, rücksichtslose Skateboardfahrer mähen Mickey über den Haufen, und bitterböse Polizisten schlagen mit dem Gummiknüpel um sich. Ihre Aufgabe besteht darin, Mickey von links nach rechts über die Straße zu steuern und allen wilden Attacken der bösen Spießbürger auszuweichen. Wehren können Sie sich nicht. Treffen die Passanten Mickey mit Nudelhölzern oder anderen neckischen Gegenständen, verlieren Sie Energie in Form von kleinen Spraydosen, die am rechten unteren Bildschirmrand zu sehen sind. Läuft ein wilder Spießbürger Sie über den Haufen, verlieren Sie gar ein Bildschirmleben. Also aufpassen! Haben Sie das Ende einer Straße erreicht, wechselt das Szenario. Jetzt wird nicht mehr mit dem Joystick, sondern mit der Maus gesteuert. Per Maus bewegen Sie eine kleine Spraydose über den Bildschirm. Sprühen Sie das einfarbige Graphity nach, das der ST Ihnen vorgibt, denn das bringt Punkte. Ist der Farbvorrat verbraucht, geht es wieder über eine Straße voller Gefahren. Natürlich sehen Straße und Spießbürger im jedem Level anders aus. Graphity Man ragt aus dem Gros der Computerspiele durch eine originelle Idee heraus. Mit einer flüssigeren Animation, einer weniger zähen Steuerung, einem hörenswerten Sound während des Spiels, mehr Möglichkeiten für den Spieler in den Straßenszenen zu agieren, wäre Graphity Man ein Hit. So bleibt es nur schwach durchschnittliche Softwarekost.

CBO



Tetra Quest



Die ersten Galaktischen Spiele stehen kurz vor der Eröffnung. Athleten aus den entferntesten Winkeln der Galaxis treffen sich in einer Arena, um Medaillen zu gewinnen. Die Fahnen wehen im Wind, und es erklingen Lieder von Einigkeit und Freundschaft. In diesem mit Weltraumbewohnern überfüllten Stadion ist Phoebus, der Sonnengott, im Begriff, das olympische Feuer anzuzünden... Wer nach Lektüre der Hintergrundgeschichte zu TETRA QUEST von MICRODEAL an ein intergalaktisches Sportspiel denkt, irrt gewaltig. Bei Tetra Quest handelt es sich nämlich um ein Ballerspiel. Wie kommen die Autoren der Anleitung von der sportlichen Einleitung zum Ballern? Ganz einfach: Eine Weltraumnation, die zu den olympischen Spielen nicht eingeladen wurde, hat aus Rache die sechs Phoenix-Tafeln gestohlen. Ohne die kann, keiner weiß warum, das olympische Feuer nicht angezündet werden. Nun sitzen Tausende von Zuschauern auf den Tribünen und warten, daß jemand die Tafeln zurückbringt. Sie können sich schon denken, wer mit dem Joystick in der Hand einen Raumgleiter über Plattformen steuern soll, um die Tafeln wiederzufinden. Stellen Sie sich Ihre Aufgabe nicht zu leicht vor. Die rachsüchtigen Diebe haben die Tafeln in 64 Einzelteile zerlegt und auf 384 Plattformen verstreut. Sie bewegen den Raumgleiter auf Schienen über die Plattformen. Durch Berührung einiger Schalter sorgen Sie dafür, daß sich Weichen umschalten, damit Sie den gewünschten Weg fortsetzen können. Teleportstationen katapultieren den Raumgleiter auf eine andere Plattform. natürlich gibt es auch eine Menge zum Abschießen. Erreichen Sie mit dem Raumgleiter den Bildschirmrand, scrollt dieser

um einen ganzen Schirm nach oben oder unten, so daß Sie die nächste Plattform absuchen können. Durch einfaches Berühren sammelt der Raumgleiter die Tafeln ein. Wurden alle Teile zusammengesetzt, sind Sie Held der Galaxis. Das Spielprinzip von TETRA QUEST bringt Abwechslung in den dichten Wald der Ballerspiele. Über Schienen gleiten, Weichen stellen, teleportieren, sammeln und ballern ist in den ersten Spielminuten noch recht amüsant, doch leider stellt sich schon nach wenigen Spielen gähnende Langeweile ein. Die grafische Darstellung weist ebenfalls Mängel auf. Beim Dauerfeuer sieht man plötzlich keine Schüsse mehr, sondern einen ununterbrochenen Strahl aus der Bugkanone des Raumgleiters kommen. Die Darstellung der Plattformen und der Gegner ist für meinen Geschmack viel zu schlicht und eintönig. Vom Sound kann ich gar nicht sprechen, ohne mir die Ohren zuhalten zu müssen. Die Zuschauer im Stadion bekommen ihr Eintrittsgeld bestimmt zurück, wenn der Held es nicht schafft, die Phoenixtafeln zu finden. Ob dem armen Käufer von TETRA QUEST ebenfalls das Geld erstattet wird, ist allerdings fraglich. Deshalb sollten Sie lieber die Finger von dem Programm lassen.

CBO

TETRA QUEST
Kategorie: Ballerspiel
Besonderheiten:
Spieler: 1
Monitor: color
Steuerung:
Hersteller: Microdeal
Info bei: Leisuresoft
Wertung:

Chronoquest
Abenteuer in Raum und Zeit



Psygnosis versucht jetzt im Grafik-Adventure-Markt Fuß zu fassen. Dem ersten Spiel "Chronoquest" liegt eine klassische Story zugrunde. Chronoquest spielt zu Beginn in einem alten Schloß im Jahre 1920. Dort ist eine Zeitmaschine versteckt - die letzte Erfindung vom Vater des Spielers. Der Vater wurde ermordet, Hauptverdächtiger ist natürlich der Spieler selbst. Aus einem hinterlassenen Brief geht allerdings hervor, daß als Mörder nur der untreue Diener Richard in Frage kommt. Und der ist mit der Zeitmaschine in die Zukunft geflüchtet. Ein Mörder der in die Zukunft geflüchtet ist - das glaubt kein Richter, also muß der Spieler selbst durch Raum und Zeit reisen, um entlastendes Beweismaterial herbeizuschaffen. Glücklicherweise hat die Maschine einen Defekt - sie kehrt zwanzig Minuten nach Reiseankunft in einer anderen Zeit stets automatisch zum Reiseausgangspunkt zurück. Richard ist also in der Zukunft, die Maschine wieder im Schloß - bloß wo? Nach dem Spielstart des vier Disketten umfassenden Chronoquest findet man sich in der Eingangshalle des Schlosses wieder. Sofort fallen die fantastischen Grafiken des neuen Psygnosis-Spiels auf - mehr läßt sich aus dem ST kaum noch rausholen. Untermalt wird das Ganze mit Animation und gelungener Musik, die sich leider ständig wiederholt und so an den Nerven des Spielers nagt.

Chronoquest wird ausschließlich per Maus gespielt. Am rechten Bildrand befindet sich eine Menüleiste mit Icons, wo man Funktionen anwählen kann. Leider sind nur sehr wenige Kommandos möglich: etwas nehmen/verlieren, öffnen/schließen, untersuchen und sich bewegen.

Jetzt also in die Zukunft reisen, Beweisstücke holen und gewonnen - Irrtum! Bis man in die Zukunft gelangt, muß man erst einmal Zeitreisen in die Steinzeit, Indien (1605 AD), Mexico (750 AD) und nach Ägypten (1100 BC) überleben.

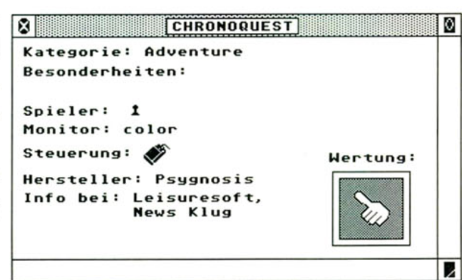
Hier zeigt, sich wie komplex Chronoquest ist. Um das Adventure zu lösen, muß man sehr viel Zeit investieren. Trotz der tollen Story und der brillanten Grafiken hat das Game mehrere Schwachstellen, die hier nicht verschwiegen werden sollen.

In Räumen kann man beispielsweise nur einen Gegenstand ablegen. Das ist oft notwendig, da man maximal 11 Sachen tragen kann. Ein abgelegter Gegenstand ist allerdings nicht im Bild zu sehen und wird auch nicht textlich erwähnt. Legt man einen weiteren Gegenstand im gleichen Raum ab, verschwindet er und taucht nie mehr auf, was sehr ärgerlich ist. Die Grafiken sind zwar Spitze, jedoch oft nur wenig aufschlußreich.

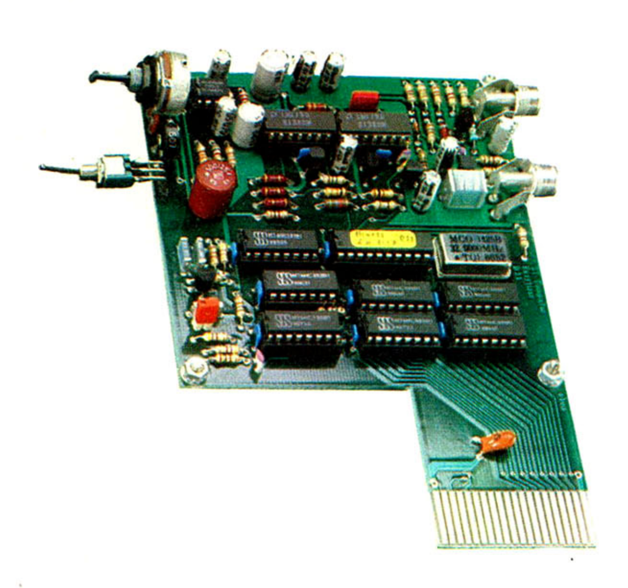
Auf der Verpackungsrückseite von Chronoquest befindet sich die Spielbeschreibung in Englisch und Deutsch; Spiel und Anleitungsheft sind allerdings komplett in Englisch. Man sollte den englischen Brief vom Vater in der Spielanleitung unbedingt lesen - notfalls mit einem Wörterbuch bewaffnet. Er enthält wichtige Informationen.

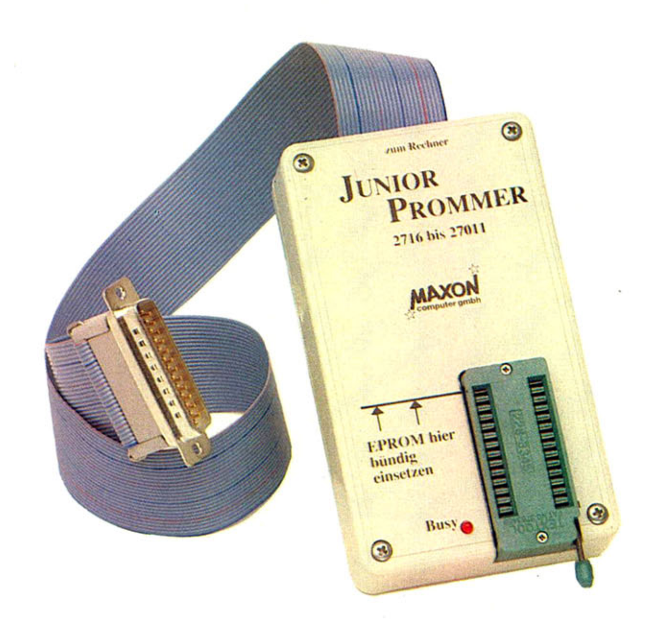
Laut Psygnosis soll mit Chronoquest eine neue Generation von Abenteuerspielen eingeläutet werden. Grafik-Adventures mit Maussteuerung gibt es jedoch schon lange (zum Beispiel Borrowed Times und Tass Times in Tone Town). In puncto Bedienungsfreundlichkeit kann Chronoquest mit den Oldies nicht mithalten. Wer hervorragende Grafiken mag und bei den erwähnten Schwachpunkten ein Auge zudrückt, findet mit Chronoquest ein storymäßig gelungenes Adventure, bei dem viele Stunden spannende Unterhaltung garantiert sind.

CBO



Aus unserer Hardwareküche





Easytizer - der Videodigitizer ohne Geheimnisse

Mit dem Easytizer können Sie beliebige Videosignale von einer Schwarzweiß- oder Farb-Kamera, Videorecorder oder direkt vom Fernsehgerät (mit Composite Video Ausgang) digitalisieren und somit auf dem Bildschirm Ihres ATARI ST sichtbar machen.
Der Easytizer wird am Modul-Port des ST angeschlossen. Die Auflösung beträgt 800x600 Bildpunkte, so daß in mittlerer Auflösung 640x200 Bildpunkte in vier Graustufen dargestellt werden können. In dieser Betriebsart werden 12,5 Bilder in der Sekunde wiedergegeben. Im hochaufgelösten Modus werden 640x400 Bildpunkte in Schwarzweiß wiedergegeben.

Besondere Merkmale des Easytizers:

- Software vollständig in Assembler
- Abspeichern der Bilder im DEGAS-Format
- Von STAD und Sympatic Paint ansteuerbar
- Ein beliebiger Bildausschnitt kann in ein mit der Maus wählbares Format vergrößert und verkleinert werden
- Spiegeln eines Bildes in horizontaler und vertikaler Ebene
- Animation mit beliebig vielen Bildern möglich, nur durch die Kapazität des Rechners begrenzt (beim Mega ST4 über 100 Bilder)
- Eingebauter Druckertreiber für die mittlere Auflösung für NEC P6/P7 und EPSON oder Kompatible
- Wahlweise automatische oder manuelle Helligkeitseinstellung
- Schnappschuß

Lieferumfang:

1) Fertiggerät
komplett aufgebaut und geprüft, inclusive Diskette mit der Easytizer Software und Bedienungsan lei tung

2) Teilsatz
Für Bastler liefern wir einen Teilsatz bestehend aus:
-Doppelseitiger, elektronisch geprüfter Platine mit Lötstoplack und Bestückungskungsaufdruck sowie vergoldeten Anschlußkontakten
- fertig programmiertes GAL 16V8
- Quarzoszillatormodul 32 MHz
- Diskette undBedienungsanleitung.

Klein, kompakt und leistungsstark- der Junior Prommer

Der Junior Prommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 KByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMs, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren. Zum Betrieb benötigt der Junior Prommer nur +5 Volt, die am Joystick-Port Ihres ATARI ST abgenommen werden, alle anderen Spannungen erzeugt die Elektronik des Junior Prommers. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen.
Selbstverständlich läßt sich ein 16-Bit Word in ein High- und Low-Byte zerlegen. Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex/ASCII-Monitor läßt sich der Inhalt eines EPROMs blitzschnell durchsuchen oder auch ändern.

Alles dabei!

Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertiggerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlußfertig geliefert. Auf der Diskette mit der Treibersoftware befinden sich noch RAM-Disk und ein Programm zum Erstellen von EPROM-Karten, ferner wird der Source-Code für Lese- bzw. Programmierroutinen mitgeliefert und last but not least ist im Bedienungshandbuch der Schaltplan abgedruckt.



Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Hiermit bestelle ich:

Name: ____________
Vornome: ____________
Straße: ____________
Ort: ____________
Unterschrift: ____________

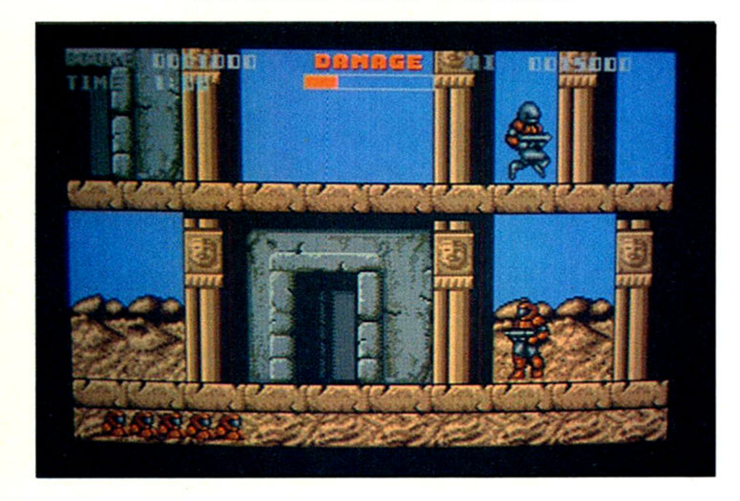
☐ Junior Prommer (Fertiggerät wie beschrieben) nur DM 189,00
☐ Leerplatine und Software (o. Bauteile) DM 49,00
☐ Leergehäuse (gebohrt und bedruckt) DM 39,90
☐ ROM-Karte 128 KByte bietet maximal 4 EPROMS Platz (fertigbestückt o. EPROMs) DM 58,00
☐ Easytizer (Fertiggerät) DM 289,00
☐ Easytizer (Teilsatz wie oben beschrieben) DM 129,00

Versandkosten: DM 7,50
Nachnahme zugzl. DM 3,50 Nachnahmegebühr.

☐ Vorauskasse
☐ Nachnahme

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrtieb in Holland: JOTKA COMPUTING Postbus 8183 NL-6710 AB Ede Tel.: 08380/38731 Fax.: 08380/21675

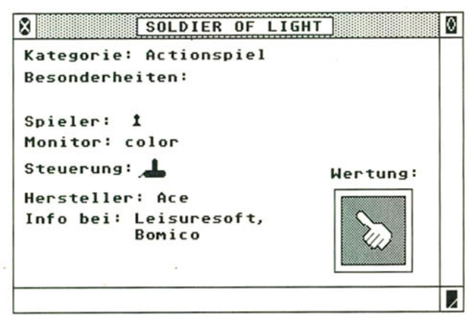
Soldat des Lichts



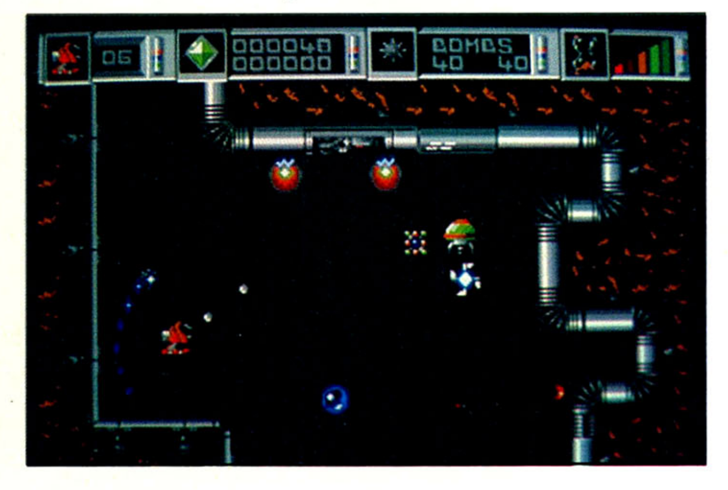
Eindringlinge aus einer anderen Galaxis haben drei Planeten unserer Milchstraße überfallen und ein ganzes Waffenarsenal abgeladen, um für einen Rückeroberungsversuch gewappnet zu sein. Hier hilft nur einer: Xain, der Superheld, der ausgerüstet mit Jetstiefeln und einer Laserkanone die Eindringlinge vertreiben soll. Per Joystick stürzen Sie sich ins Abenteuer. Zunächst erscheint eine Art Menü auf dem Bildschirm. Mit einem Raumschiff sollen Sie den Planeten anfliegen, den Sie als erstes zu säubern gedenken. Kaum hat man sich einen Planeten ausgesucht, geht es auch schon los: Ein Raumschiff schwebt über der Planetenoberfläche und Xain springt heraus. Sofort greifen die feindlichen Soldaten an. Xain läuft bei horizontalem Scrolling (ruckelt etwas!) über die Planetenlandschaft und ballert mit seiner Laserkanone auf alles, was sich bewegt. Haben einige der feindlichen Soldaten ein Leben ausgehaucht, hinterlassen sie Extrawaffen, die Xain nur noch einzusammeln braucht. Dadurch erhöht sich seine Feuerkraft. Mit Hilfe seiner Jetstiefel vermag es Xain, in ein anderes Stockwerk oder auf in der Luft fliegende Felsen zu schweben. Sind Sie durch die gesamte Landschaft eines Planeten gelaufen, wartet ein gefährliches Monster auf den Helden. Sollte es Ihnen gelingen, ein Monster zu besiegen, gibt es eine kleine Shoot'em up-Szene, in der Sie ein Raumschiff bei horizontalem Scrolling durch den Weltraum fliegen.

Dann geht es auf den nächsten Planeten. Wieder gibt es Extrawaffen, Soldaten und ein Monster am Ende des Spielabschnittes, bevor Sie mit dem Raumschiff zum nächsten Planeten fliegen. Hat man auch diesen durchgespielt, muß sich der Spieler beim Todesstern beweisen. Wie es da aussieht, kann ich Ihnen leider nicht beschreiben. Nach mehreren Tagen intensiver Tests hatte ich immer noch Schwierigkeiten, mich über den letzten Planeten zu kämpfen. (Wer das Spiel durchspielt, kann mir ja mal schreiben!!!) SOLDIER OF LIGHT bietet lange Unterhaltung. Manchmal sind die Angriffe auf unseren Helden ein bißchen heftig, was unweigerlich zum Verlust eines Bildschirmlebens führt. Doch mit etwas Übung entkommt man auch aus noch so aussichtslos erscheinenden Situationen. Wunderschöne Grafiken und nervenaufreibende Szenen werden auch Sie für lange Zeit in Atem halten.

CBO

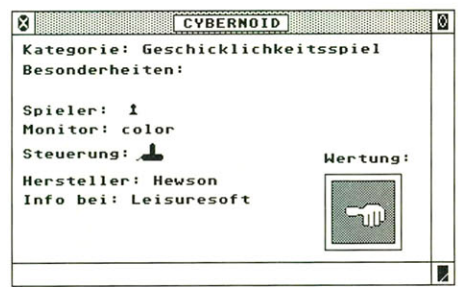


Cybernoid



Bitterböse Piraten haben die Laderäume der Föderation geplündert und dabei Mineralien, Edelsteine und militärisches Gerät mitgehen lassen. So geht das nicht! Den Schurken muß unbedingt das Handwerk gelegt werden. Raten Sie, wessen Aufgabe es sein wird, mit einem Superraumschiff loszudüsen und in einem Labyrinth die gestohlenen Güter wieder zu beschaffen? Keine Panik! Das Superraumschiff ist bis an die Scheinwerfer bewaffnet. Außerdem warten im Labyrinth noch einige Extrawaffen darauf, eingesammelt zu werden: Bomben, Minen, Schutzschilder, Suchraketen und Sprungbomben befinden sich schon an Bord des Raumschiffes. Alle Waffen lassen sich per Funktionstasten auswählen und mit dem Feuerknopf aktivieren. Sie brauchen jede einzige, da Ihr Kampfschiff ständig von feindlichen Raumschiffen umgeben ist. Im Labyrinth haben die bösen Piraten auch allerlei neckische Hindernisse aufgestellt. Da gibt es Schnellfeuerkanonen und Raketenwerfer, die, in einem tiefen Vulkan installiert, unentwegt auf des Spielers Raumschiff feuern. Dabei gilt es, die richtige Waffe auszuwählen, um das Hindernis beseitigen zu können. Auf der Reise durch das Labyrinth sehen Sie stets einen Screen, der umschaltet, wenn Sie den Bildschirmrand erreichen. Unter Zeitdruck heißt es, das Depot der Piraten zu erreichen. Irgendwo im großen Labyrinth befindet sich das Lager der Piratenbrut. Hier finden Sie alle gestohlenen Gegenstände. . Die Grafik von "Hewsons" "Cybernoid" liegt weit über dem Durchschnitt. Flüssige Spriteanimation, farbenfrohe Zeichnungen und die Explosionen der feindlichen Kanonen verstehen zu begeistern. Bei der Bewertung des Sounds fiel nur die Titelmelodie positiv auf. Dafür bietet "Cybernoid" eine Menge Spielspaß. Man muß in einigen Situationen schon eine große Portion Geschick aufweisen, um weiterzukommen.

CBO



Leben und sterben lassen



James Bond 007, die absolute Doppelnull, hüpft mal wieder von der Leinwand auf den Bildschirm. Nach den mißlungenen Bondspielen "A view to a kill" und "Der Hauch des Todes" schmeißt DOMARK mit "Leben und sterben lassen" (diesmal in einer Gemeinschaftsproduktion mit ELITE) schon das dritte Bond-Game auf den Markt. Nach dem dritten Anlauf ist Domark jetzt endlich der große Wurf gelungen. Bei "Leben und Sterben lassen" ist die Hintergrundgeschichte ist zwar genauso hohl wie die Handlung in den Filmen, aber darüber können wir ja hinwegsehen.

Dr. Kananga, ein rücksichtsloser Premierminister einer karibischen Insel, besitzt riesige Mohnfelder. Tonnenweise bringt er das daraus gewonnene Heroin auf den amerikanischen Markt. Das Schlimme ist, er läßt es dort kostenlos verteilen, damit möglichst viele Leute davon abhängig werden und dann später für teures Geld Heroin bei ihm kaufen müssen. Mr. Big, wie der niederträchtige Minister in Unterweltskreisen auch genannt wird, verschifft das Heroin von einem geheimen Hafen einer Karibikinsel. Klar, daß Bond nicht mit dem Treiben des windigen Geschäftemachers einverstanden sein kann. Er wird von seinem Boss "M" ins Feld geschickt, um dem bösen Treiben ein Ende zu bereiten. Dazu muß er mit einem Schnellboot durch eine Wasserstraße brausen und mit der Bordkanone auf die Verteidigungsanlagen des Heroinherstellers ballern. Der hält sich eine Privatarmee, um alle Eindringlinge abzuschrecken. So greifen Bond bei seiner rasanten Fahrt Flugzeuge und andere Schnellboote an. Vorsicht, im Wasser liegen explosive Fässer, Minen und natürliche Hindernisse wie Felsen und Eisblöcke. Die Hindernisse, die der Feind in den Weg stellt, sind aber nicht Bonds einzige Sorge. Sein Motorboot frißt Sprit wie ein Esel Hafer. Deshalb muß er darauf achten, im Wasser herumschwimmende Benzinfässer einzusammeln. Auf einer Anzeige im Cockpit sieht man, wie es um den Tankinhalt bestellt ist.. Das Durchfahren von Tunneln und Schleusen macht die Motorbootfahrt sehr realistisch und vor allem schwierig. Damit sich der Spieler an sein Motorboot und die Verteidigungsanlagen gewöhnen kann, hat Domark an drei Trainingsmissionen gedacht. In der eigentlichen Mission muß man die Wasserstraße so weit wie möglich durchfahren, um am Ende des Flusses auf die heroinverarbeitenden Fabriken von Mr. Big zu stoßen, die Sie mit einem gezielten Raketenschuß zerstören müssen. Fazit: "Leben und Sterben lassen" ist ein erstklassiges Game mit hervorragender Grafik, einem fetzigen Titelsound und einem süchtig machenden Spielprinzip.

CBO



LEBEN UND STERBEN LASSEN

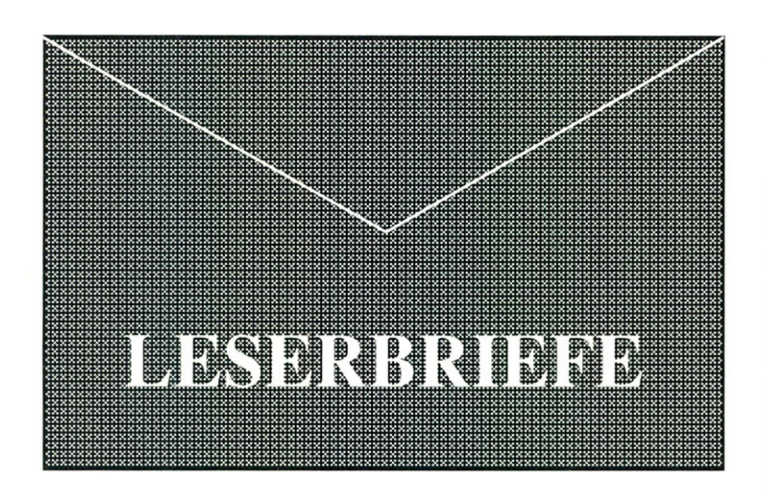
Kategorie: Motorbootrennen
Besonderheiten: sehr schnelle Grafik

Spieler: 1
Monitor: color
Steuerung:

Hersteller: Domark / Elite
Info bei: Leisuresoft, Bomico

Wertung:

ENDE



Spracherkennung

Bei Planungen zu einer Dissertation innerhalb der Sozialwissenschaften stehe ich (und viele Kollegen und Kolleginnen) vor dem Problem, größere Mengen an Tonbandprotokollen von Interviews verschriftlichen zu müssen.

Ist es möglich, diese Aufgaben dem Computer zu überlassen?

Wenn ja, mit welchem Aufwand an Soft- und Hardware. So einfach wird es sicher nicht gehen, daß man den Kassettenrekorder an die MIDI-Schnittstelle hängt und ein ASCII-Text auf Diskette hinterher findet (vorhandene Hardware: ATARI 1040STF).

(Rudolf Schmitt, Berlin)

Red.: Leider ist dieses Problem lange nicht so einfach zu lösen, wie man vielleicht glauben mag. Eine Vielzahl von Wissenschaftler rund um den Erdball forschen danach, wie man am geschicktesten Sprache erkennen kann. Dabei sind Teilerfolge erzielt worden, bei denen ein begrentzter Wortschatz von diversen Rechnern schon erkannt werden können, wobei häufig aber die Einschränkung gilt, daß die Worte von einundderselben Person und dazu noch relativ gleichmäßig und deutlich gesprochen werden müssen, da größere Abweichungen durch Klangänderungen vom Rechner nicht mehr als gleich erkannt werden. Schaut man sich das Signal des Buchstabens *'a'* von zwei verschiedenen Personen an, so können die Unterschiede sehr groß sein. Die Wissenschaft forscht nun daran, welche Merkmale bestimmte Vokale oder Laute auszeichnet. Aus der schwierigen Aufgabenstellung erkennt man sofort, daß weder der ST noch die bisherige bekannte Software auf diesem Sektor auch nur annähernd dazu ausreicht, Sprache in der Praxisanwendung zu verschriftlichen. (Übrigens: An die MIDI-Schnittstelle schließt man kein normales AUDIO- (also analoges) Signal sondern ein schon in digitale Daten gewandeltes AUDIO-Signal an.) Allerdings gibt es ein Bereich, der weiter ist als die Spracherkennung: Dieser Bereich ist die Schriftenerkennung. Es gibt schon Programme (auf dem ST), die es ermöglichen, einen schon geschriebenen Text, der eventuell nur auf Papier vorliegt, mit einem sogenannten Scanner aufzunehmen und dann durch ein Programm so abzutasten, das mit hoher Wahrscheinlichkeit die Grafik erkennt und in einen normalen Text wandelt. Ein solches Programm wurde auf der ATARI-Messe unter dem Namen AUGUR vorgestellt - sicherlich werden wir es in einer der nächsten Ausgaben ausführlich vorstellen.

*

Ausblendung von Druckerzeichen bei 1st_Word

Die Ausblendung von Drukkerzeichen auf dem Bildschirm läßt sich auch in selbstgestrickten HEX-Files problemlos erreichen: Man braucht nur in der Konvertierungstabelle hinter die HEX-Nummer des zu unterdrückenden Zeichens ein Sternchen (und sonst nichts!) einzutragen, also zum Beispiel *"BA*"*. Dann kann dieses Zeichen auch auf dem Bildschirmfenster nicht angeklickt werden.

(Konstantin Diederichs, Ottobrunn)

Red.: Vielen Dank für diesen Tip. An dieser Stelle möchten wir uns einmal bei allen Lesern bedanken, die uns immer wieder Tips zu 1st_Word einsenden! Langsam könnten wir fast Serie daraus gestalten. Monat für Monat wird immer wieder gezeigt, daß viel mehr Features in 1st_Word stecken als allgemein bekannt ist. Also nur Mut beim Einsenden von Tips, sie sind gerne willkommen .... im Folgenden wollen wir aber kurz zwei Mängel von 1st_Word Plus veröffentlichen, um Anwender davor 'zu warnen':

*

Mängel und deren Bewältigung an 1st_Word Plus

Seit Anfang August unterrichte ich meine Schüler in der Anwendung von 1st_Word Plus. Ich selbst arbeite schon seit fast zwei Jahren mit dem Programm und bin im Großen und Ganzen damit zu frieden. Die in den Fachzeitschriften genannten Verbesserungsvorschläge kann ich zwar auch unterstreichen, aber insgesamt läßt sich mit dem Programm sehr gut arbeiten. An der neuen Version, die meine Schule nun bekommen hat, habe ich jedoch zwei ganz erhebliche Mängel festgestellt, die nach meiner Meinung abgestellt werden müßten:

1. Das Wörterbuch lade ich von einer Extra-Diskette. Dadurch kann es vorkommen, daß ich vergesse, die Diskette zu wechseln, wenn ich *'Speichern & Weiter'* aufrufe. Das Programm registriert dann nicht, wenn man anschließend die Diskette wechselt, sondern bringt immer wieder die Warnbox *'Schreib/Lesefehler!'*. Bisher konnte ich mir in solchen Fällen damit helfen, daß ich einfach eine andere Datei aufrief und dann *'Abbruch'* anklickte. Bei der neuen Version bleibt dann jedoch die Datei-Auswahl-Box auf dem Bildschirm und der Text läßt sich nicht mehr weiterverarbeiten. Es hilft nur noch *'Speichern als...'* und der Neuaufruf der Datei, was bei längeren Dateien natürlich sehr lästig ist.

2. Wenn man aus dem Menü *'Korrektur'* versehentlich die Option *'Korrekturende'* anstatt *'Wort hinzufügen...'* erwischt, gibt es kein Zurück mehr. Man ist dann zum Abspeichern des Wörterbuches gezwungen, denn mit *'Abbruch'* geht die Erweiterung des Wörterbuches verloren. Anschließend muß man die ganze Prozedur des Wörterbucheinladens wiederholen.

(Heiko Müller, Edewecht)

*

Talkshow

Ich muß ein Programm schreiben, mit dem man sämtliche Tasten- und Mausaktionen eines beliebigen Programmes aufzeichnen und danach wieder zeitlich richtig abspielen kann. Dadurch sollte es möglich sein, zum Beispiel eine Installation mit einem bestehenden Programm zu automatisieren, ohne daß man irgendeine Programmänderung vornehmen muß. Verwenden wollte ich dazu die beiden AES-Funktion *APPL_TRECORD* und *APPL_TPLAY*. In der ST-Computer 4/88 habe ich glücklicherweise die ST-

Ecke *'Und sie dreht sich doch....'* gefunden, welcher eigentlich auf die Schnelle betrachtet genau mein Problem lösen sollte. Nach genauerem Studium des Artikels ist bei mir aber die Frage aufgetreten, wie man denn nun konkret mit einem Aufzeichnungsprogramm von irgendeinem x-beliebigen Programm die Maus- und Tastenereignisse aufzeichen kann. Danach sollte es möglich sein, mittels Abspielfunktion alle zuvor aufgezeichneten Daten zeitlich richtig an das Programm zurückzugeben und dieses somit zu steuern. Dies bedeutet ja, daß zwei Programme gleichzeitig aktiv sind, wobei das eine arbeitet wie normal und das andere dessen Aktionen aufzeichnet und irgendwo abspeichert.

Wie löst man dieses Problem?

(Rene Hanslin, ATARI Schweiz AG)

Red.: Zunächst einmal möchte ich Ihren Eifer leider etwas dämpfen, denn so einfach mit *Appl_trecord* und *Appl_tplay* geht es nicht, weil für solche komplexe Aufgaben, die sogar über andere Programme als das eigene hinausgehen, sie nicht gedacht sind. Tests, die aus einer Accessory, bei gleichzeitigem Vorhandenseins des Desktops, Daten übermittelten, ergaben folgendes Ergebnis: Die Steuerung der Maus funktionierte noch einwandfrei, aber ein Klick oder gar Doppelklick kamen nicht zur Ausführung. Folgerung: Obwohl eine Accessory quasiparallel zur aktuellen Applikation läuft, ist es von ihr aus nicht möglich, Befehle mit *appl_tplay* korrekt zu übermitteln. Meines Erachtens ist die ganze obige Problemstellung nur auf unterster Ebene (Betriebssystemebene) zu lösen, in dem man zunächst alle externen Eingaben wie Maus und Tastatur zeitgerecht aufzeichnet. Dies kann man bespielsweise dadurch erreichen, daß man sich in den *IKBDSYS*-Vektor einhängt, der alle Befehlspakete des Tastaturprozessors abfängt und versehen mit einer Zeitinformation abspeichert. Beachten Sie aber dabei, daß dies alles im Interrupt passiert und ein tatsächliches Abspeichern auf Diskette von hier aus nicht möglich ist! Vielmehr müßte dies zum Beispiel durch eine Accessory geschehen, die Zugriff auf diesen Puffer hat. Beim Abspielen müßten die Daten dann aufgrund der Zeitinformation in IKBDSYS eingespielt werden. Trotzdem möchte ich darauf hinweisen, daß gerade Tastaturroutinen sehr sauber programmiert sein müssen und auch aufgrund der Interrupt-Struktur sehr schwer zu debuggen sind. Abgesehen davon kommen pro Sekunde eine Menge an Eingabeinformationen an, wobei in dem oben erwähntem Beispiel das ganze Tastaturprozessorpaket abgespeichert werden muß. Eine Realisierung dieser Ideen liegen aufgrund von Zeitmangel und Komplexität von meiner Seite noch nicht vor. Vielleicht hat ein Leser diese Ideen oder ein ähnliches Konzept schon programmiert. Andernfalls möchte ich hiermit alle Assemblerfreaks aufrufen, dieses Problem vielleicht einmal in die Tat umzusetzen.

*

Programmheader

Ich möchte an käuflich erworbenen Programmen Änderungen (z. B. sinnlose Alertboxen bei Programmende entfernen) vornehmen und sie dazu disassemblieren. Zwar findet man in der Literatur Informationen über den Aufbau der *Base-Page*, gibt es aber Unterlagen, welche Information sich auf der jeweiligen Diskettendatei vor dem eigentlichen Programmcode befinden? Ich möchte die Programme nicht mittels *PEXEC* in den Speicher laden, da dann Speicherplatz für unter Umständen große *BSS*-Segmente benötigt wird.

(Jürgen Horzitz, Gießen)

```
typedef struct
{
    int ph_branch;  /* Sprung an den Anfang des Programmsegments:*/
                    /* immer 0x601A */
    long ph_tlen;   /* Länge des Text-Segments */
    long ph_dlen;   /* Länge des Daten-Segmentes */
    long ph_blen;   /* Länge des BSS-Segmentes */
    long ph_slen;   /* Länge der Symboltabelle */
    long ph_res1;   /* reserviert */
    long ph_res2;   /* reserviert */
    long ph_flag,   /* nach Originaldokument auch reserviert, aller-
                       dings bewirkt ein Setzen dieses Eintrages ein
                       Nicht-Relozieren des Programms und ein Nicht-
                       Löschen des BSS-Segmentes */
}
```

Listing 1

Red.: Ob nun Alertboxen am Programmende wichtig sind oder nicht, soll dahingestellt bleiben, aber es gibt Informationen über den Programmheader, die wir veröffentlichen können. Das Programmformat des ST unterscheidet sich von Formaten wie die des C-64 oder des 8-Bit-ATARI, da die Programme an jeder beliebigen Stelle im Speicher ablauffähig sein müssen und daher vom Betriebssystem noch einmal 'in die Mangel' genommen werden. Diese Informationen sind im Datei-Kopf, in den *TEXT-*, *BSS-* und *DATA-*Abschnitten und der optionalen *Symbol-* und *Relozierta-belle* vorhanden. Das Format des Datei-Headers ist in Listing 1 beschrieben.

Der erste Eintrag ist ein Assemblercode des 68000, der bewirkt, daß einfach hinter den Header zum Programmstart gesprungen wird. Die anderen Einträge geben die Länge der unterschiedlichen Programmteile an. Normalerweise sollten die drei restlichen Bytes nach der Original Digital-Research-Dokumentation reserviert sein, allerdings wird der dritte Eintrag doch genutzt, wie man im *Sonderheft 2/ST Computer* in *TOS-Intern* von Alex Esser lesen kann. Ist der Eintrag gesetzt, so wird das Programm nicht reloziert und das BSS-Segment nicht gelöscht. Wahrscheinlich sind das Anlehnungen an MS-DOS, bei denen es schon mal aus Geschwindigkeitsgründen vorkommt, daß solche Maßnahmen ergriffen werden und auf feste Adressen basierende Programme schneller zugegriffen werden kann. Ich möchte Sie auch darauf hinweisen, daß es zum Disassemblieren von Programmen schon Programme auf dem Markt gibt, die sogar Symbole unterstützen und den Code assemblierfähig abspeichern. Aus Erfahrung kann man sagen, daß ein Arbeiten mit solchen Programmen, die in letzter Zeit vermehrt angeboten werden, sehr gut möglich ist

*

PC ditto & die VORTEX-Festplatte

Seit Juni 1988 arbeite ich immer wieder mit *PC ditto* auf einer *VORTEX HDplus 20*-Festplatte mit zwei Partitionen auf Partition D ohne Probleme! Der "Trick" liegt darin, das *AHDFIX.PRG* aus dem STPROGRA-Order von PC ditto auch auf die VORTEX Platte anzuwenden (vorsichtshalber habe ich ein Backup der Platte angelegt - man weiß ja nie !!!). Das AHDFIX.Prg ist im AHDFIX.DOC gut dokumentiert, so daß eine Erklärung hier entfallen kann. Eigentlich müßte dieser Tip auf alle VORTEX plus -Platten anzuwenden sein, da der Treiber wohl derselbe ist (?!).

(Martin Sauer, Schwabach)

*

ST-Erweiterungen

Ist es möglich, abgesehen vom Gehäuseplatz, am 1040STF weitere Platinen wie beim MEGA-ST anzuschließen, wie zum Beispiel der Arithmetik-Coprozessor und ähnliche Erweiterungsplatinen (Graphikkarte, IEEE-Karte)?

(Hartmut Broich)

Red.: Das Anschließen von Erweiterungskarten am 1040-ST(F) ist leider nicht so einfach möglich wie beim MEGA-ST, was daran liegt, daß der gesamte Bus des 68000 beim MEGA ST auf eine gemeinsame Leiste geführt worden ist. Prinzipiell ist natürlich möglich, Erweiterungen am 1040 anzuschließen, wenn man sich die entsprechenden Leitungen vom Prozessor zur Karte zieht, wobei dies natürlich mit ein wenig Arbeit und Bastelei verbunden ist. Für den Blitter gibt es übrigens schon eine Adapterplatine für den 1040, die wir in der *ST-Computer 11/88* vorgestellt haben.

*

Probleme mit dem Icon

Ich habe Probleme, in C das Zeichen eines Icons zu verändern: Dabei habe ich im Programm folgende Zeile geschrieben:

((ICONBLK) baum [DEF].ob_spec) ->ib_char = 65;

Der Icontext läßt sich problemlos verändern, aber wenn man versucht, das Zeichen nach dieser Methode zu ändern, compiliert der Compiler zwar fehlerfrei, aber alle Icons sind beim Programmdurchlauf weiß. Hierzu ist noch zu sagen, daß ich nicht etwa mit *long*, sondern mit *OBJECT-Strukturen* gearbeitet habe.

Wie kann man das fehlerfrei beschreiben ?

(Christopher Knierim, Hagen)

Red.: Erstmal ist es sehr richtig und gut, mit Objekt-Strukturen zu arbeiten, da man ja nie weiß, ob irgendwann die Struktur mal verändert wird - dann wäre nur eine neue Compilierung nötig. Abgesehen davon liegt Ihr Problem an folgendem: Entgegen der weitläufigen Meinung beinhaltet das Strukturelement *ib_char* des Iconblocks nicht nur den Buchstaben, sondern auch die Farbe des Icons. Diese Farbe wird nochmal in die Farbe der Maske und des Vordergrundes unterteilt. Sie werden in vier Bits kodiert, wobei der Vordergrund in den oberen und der Hintergrund (die Maske) in den unteren vier Bits des Highbytes von *ib_char* kodiert sind; das untere Byte enthält den ASCII-Code des gewünschten Zeichens. Damit ist auch klar, was Sie falsch gemacht haben: Sie haben schlichtweg die beiden Farben auf Null gesetzt! Wenn Sie ein neues Zeichen setzen möchten, so müssen Sie vorher die unteren 8 Bits löschen und dann ihr neues Zeichen in *ib_char* mit einer Oder-Funktion hineinmaskieren: (siehe Listing 2)

```
((ICONBLK)baum[DEF]  .ob_spec)->ib_char &= 0xf0;
                     /*unteres Byte löschen*/
((ICONBLK)baum[DEF]  .ob_spec)->ib_char |= 65;
                     /* Zeichen setzen */
```

Listing 2

Auf diese Weise läßt sich das Zeichen sauber setzen. Auf die gleiche Art kann man natürlich auch die Farben des Icons manipulieren.

*

KORREKTUR Genauigkeit erwünscht

In der Ausgabe *7/88* veröffentlichten wir ein kleines Programm zur Messung von Zeiten im Millisekunden-Bereich. Leider sind uns dabei zwei Fehler unterlaufen. Wir drukken das Programm hier noch einmal ab. Die Fehler befanden sich in den mit * gekennzeichneten Zeilen: (siehe Listing 3)

```
'millisec, R.Kracht, Gärtnerwiete 9, 2085 Quickborn, 29.5.88 '

Bild%=Xbios(2)              ! Adresse des Bildschirmspeichers
Zeit%=Bild%+32200           !* Adresse der Timer-A-Interrupt-
Routine Uhr%=Zeit%+20           ! Adresse der Uhr
Lpoke Zeit%,&H8B90005           ! Anfang der Interrupt-Routine
Lpoke Zeit%+4,&HFFFA0F
Lpoke Zeit%+8,&H27CF3FF
Dpoke Zeit%+12,&H52B9
Lpoke Zeit%+14,Uhr%             ! Adresse der Uhr
Dpoke Zeit%+18,&H4E73           ! rte (Ende)
Lpoke Uhr%,0                    ! Uhr auf Null stellen
A%=Xbios(31,0,2,246,L:Zeit%)    ! Uhr starten
Do
 Void Inp(2)
 U=Lpeek(Uhr%)              ! Uhr lesen
 U=U+U/1024                 ! Uhr korrigieren
 U=Int(U+0.5)               ! auf volle Millisekunden runden
 Print U;" msec seit Uhrenstart"
Loop
```

Listing 3

ENDE



DIE GEM PROFIS KOMMEN™

dBMAN inkl. Tempus-Editor und Toolbox

Hotline in Deutsch, Englisch, Holländisch, Französisch

Bundesrepublik Deutschland
COMPUTER MAI
DM 624,–
Tel.: 089/4480691
Fax: 089/4483820

Niederlande
Softpaquet International
HFL 695,–
NL 2724 GT Zoetermeer
Tel.: 0031/79/423571

Belgien
Micro-Connection
BF 13900,–
Tel.: 0032/3/2311540

Frankreich
16–32 Diffusion, Paris
Tel.: 0033/1/46622177 79

Österreich
Kneisz, GesmbH
öS 4999,-
A-1120 Wien
Tel.: 0222/552950 und 552959

Schweiz
ADAG-Computershop
SFR 598,–
CH-Zürich
Tel.: 0041/1/2521868

dBMAN by Klaus Mai ist ein eingetragenes Warenzeichen von Klaus Mai

Die ganze Kraft von dBase III plus

für Ihren Atari ST

VERSION 5.1 mit COMPILER DM 998,–

XLISP 2.0

Eine neue PD-Version für den ATARI ST

XLISP ist die bekannte PD-Version der Sprache LISP vom Altmeister David M. Betz, Senior-Editor der Zeitschrift BYTE in den USA. In der PD der ST Computer sind der Reihe nach die Versionen 1.4, 1.5 und 1.7/1.71 erschienen. Alle Versionen waren von vornherein als experimentell und damit als vorläufige Versionen gekennzeichnet. Die neueste Version 2.0 vom 6.2.1988 macht da schon eher einen beständigen Eindruck. Der folgende Bericht soll die Eigenschaften dieser XLISP-Version im Vergleich zu den vorigen untersuchen. Denn immerhin ist XLISP für den ATARI ST Computer zur Zeit die am ehesten dem Common LISP-Standard entsprechende LISP-Version und dazu noch kostenlos.

Äußeres

Auf der PD-Diskette sind ca. 256 kByte Information und Programme enthalten. Davon entfallen ca. 82 kByte allein auf die Dokumentation, die im knappen Stil der vorigen Dokumentation gehalten ist. Auch das neue XLISP 2.0 verzichtet auf GEM und kommt gleich nach dem Start im VT52-Modus zur Sache. Es werden auch wieder einige nützliche Utilities wie der Pretty Printer *PP.LSP* mitgeliefert, deren Syntax wurde allerdings geändert. Funktionen und Macros können mit *(PP-DEF <Funktionsname> <Stream>)* auf einem Stream, (Default: Bildschirm) ein File kann mit *(PP-FILE <Filename> <Stream>)* ausgegeben werden. Hier ist im Original ein Fehler. Es wird die Funktion *OPENI* verlangt, die nirgends deklariert ist. Ich habe diese Funktion im File *PP* eingefügt und hoffe, daß sie funktioniert. Auf der Diskette ist weiterhin ein Programm *QA.LSP* aufzufinden. Es handelt sich hier um die LISP-Version der Resolutionsmethode, wie sie in allen PROLOG Interpretern angewendet wird.

Änderungen am Programm

Es scheint so, daß die Änderungen am Programm im wesentlichen zur Fehlerreduktion des Interpreters geführt haben. Fehler der Version 1.7 bzw. 1.71 wie die Fehlfunktion der *PROG*-Funktion bzw. die fehlerhafte Funktion der *OBJECT*-orientierten Funktionen scheinen jetzt behoben zu sein. Der Sprachumfang ist etwas erweitert und noch mehr dem Common LISP-Standard angeglichen worden. Schauen wir uns die Änderungen im Detail an Hand des Handbuches an.

Die Unterschiede beginnen beim Start des Interpreters. Wenn XLISP 2.0 gestartet wird, wird zunächst im aktuellen Inhaltsverzeichnis nach der Datei *XLISP.WKS* gesucht, die den zuletzt gesicherten Workspace enthält. Diese kann mit der *SAVE*-Funktion abgespeichert werden und steht danach mit dem Start zur Verfügung. Man kann sich also seine eigene Version von XLISP zurechschneiden und in Zukunft den Workspace mit allen Einstellungen beim Start übernehmen. Dann erst wird *INIT.LSP* geladen und anschließend die in der Kommandozeile angegebenen Dateien.

An Datenstrukturen bietet die neue Version folgende zusätzliche Datentypen: *Characters*, *Streams* und *Closures*. *Characters* bedürfen keiner weiteren Erläuterung. *Streams* sind eigentlich nur die alten File Pointer, deren Syntax in Richtung Common LISP angeglichen wurde. Als *Closures* werden die benutzerdefinierten Funktionen bezeichnet. Völlig neu ist also eigentlich nur der Character Datentyp.

Neue Symbole sind ebenfalls hinzugekommen. Es handelt sich um **error-output**, **trace-output** und **debu-output**, die den jeweiligen Ausgabestrom definieren. Mit den Symbolen **integer-format**, **float-format** und **print-case** kann das Ausgabeformat festgelegt werden. Die Symbole '+', '++' und '+++' sind an die letzten drei Eingaben gebunden, entsprechend '*', '**' und '***' an die drei letzten Ergebnisse.

Die neuen Evaluationsfunktionen sind *(GET-LAMBDA-EXPRESSION <closure>)*, die den entsprechenden Lambdaausdruck zurückgibt; *(MACROEXPAND <form>)* bzw. *(MACROEXPAND-1 <form>)*, die den entwickelten Macroausdruck *<form>* zurückgeben.

Endlich ist im Bereich der Symbolfunktionen auch die Funktion *PSETQ* aus dem Common LISP-Reservoir definiert.

An Array-Funktionen ist *(VECTOR <expr>)* neu hinzugekommen, die einen neuen Vektor initialisiert.

In LISP stellen die Listenfunktionen ja den Kern der Sprache dar. Hier sind nun endlich die leichter verständlichen Funktionen *FIRST*, *SECOND*, *THIRD*, *FOURTH* und *REST* als Synonym für

CAR, *CADR*, *CADDR*, *CADDR* und *CDR* definiert worden. Neu sind auch *(REMOVE-IF <test> <list>)* und *(REMOVE-IF-NOT <test> <list>)*, die aus einer Liste je nach Ausgang des Tests Elemente entfernen. Die entsprechenden destruktiven Funktionen lauten *DELETE-IF* bzw. *DELETE-IF-NOT*. Als sehr nützlich dürfte sich auch der neue *SORT*-Befehl erweisen.

Auch bei den Prädikatenfunktionen sind neue hinzugekommen. *ENDP* testet das Ende einer Liste, *INTEGERP*, *FLOATP*, *STRINGP CHARACTERP*, *ARRAYP*, *STREAMP*, *OBJECTP* testen, welcher Datentyp vorliegt.

Bei *WHEN* und *UNLESS* handelt es sich um zwei neue Kontrollstrukturen, die dem *IF*-Konstrukt sehr ähnlich sind. *FLET*, *LABELS*, *MACROLET* und *UNWIND-PROTECT* erzeugen neue Funktionen bzw. kontrollieren die Evaluation eines Terms.

An Schleifenkonstrukten ist das gute alte *LOOP* hinzugekommen.

Bei den *PROGRAM*-Features ist es nun möglich, Blöcke zu benennen und abzuarbeiten.

Die Funktionen *TRACE* und *UNTRACE*, die bisher extra geladen werden mußten, sind nunmehr im Standardumfang der Sprache enthalten.

Bei den arithmetischen Funktionen ist *GCD* (der größte gemeinsame Teiler) hinzugekommen.

Dramatisch verändert hat sich die String-Bibliothek. Sie hat sich erweitert um Funktionen, die einen String trimmen (*STRING-TRIM*, *STRING-LEFT-TRIM*, *STRING-RIGHT-TRIM*), in Groß- bzw. Kleinschrift umsetzen (nichtdestruktiv: *STRING-UPCASE*, *STRING-DOWNCASE*; destruktiv: *NSTRING-UPCASE*, *NSTRING-DOWNCASE*) sowie mehrere Strings miteinander vergleichen können (was in BASIC, PASCAL und C schon lange geht).

Ganz neu sind die Character-Funktionen, die Zeichenextraktion aus einem String ermöglichen, auf das Vorliegen von Klein-/Großbuchstaben, Ziffern oder ähnlichem testen bzw. die entsprechende Umwandlung übernehmen. Auch Zeichenvergleiche sind möglich.

Die Terminologie der Input-/Output-Funktionen wurde dem *Stream*-Konzept von Common LISP angeglichen. Was bisher als File Pointer deklariert war, wird nun als *Stream* deklariert. Neu ist hier allerdings die Möglichkeit formatierter Ausgabe. Ebenso ist die Ein-/Ausgabe über String-Streams neu.

Bei den Systemfunktionen ist die Möglichkeit hinzugekommen, den aktuellen Workspace abzuspeichern und später wieder zu restaurieren.

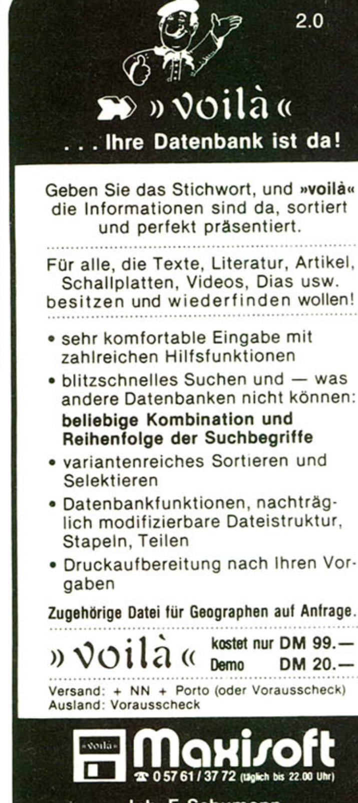
Resümee

Das XLISP 2.0 ist dem Common LISP-Standard noch näher gerückt. Es wurde mit dem Mark Williams C-Compiler compiliert. In den Quelltexten der mitgelieferten LISP-Programme sind die Änderungen von der Version 1.7 zur Version 2.0 nicht konsequent durchgeführt, die notwendigen Korrekturen (siehe *OPENI* wie oben erwähnt) dürften aber auch LISP-Anfängern leichtfallen. Es lohnt sich für alle, die sich für LISP interessieren, diese neue Version zu besorgen.

Dr. K. Sarnow

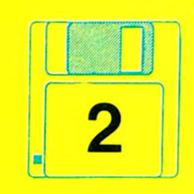
ENDE







STARKE SOFTWARE



NEUE VERSION 2.6

ST-COMPUTER Index
DAS JAHRESINHALTS-VERZEICHNIS 1986/1987

Wer sich einen schnellen Überblick darüber verschaffen möchte, welche Artikel 1986 und 1987 in **ST COMPUTER** erschienen sind, der kann jetzt mit **ST COMPUTER Index** die Jahrgänge 86/87 der Zeitschrift komfortabel als Handbuch und Nachschlagewerk nutzen.

Die Diskette enthält:
- **eine Indexdatei,** in die die Registereinteilung, zahlreiche Schlagwörter sowie Namen von Listings aufgenommen sind. Sie enthält 11 Datenfelder mit den Bezeichnungen Autor, Titel, Schlagwort, Stichwort, Index, Bemerkungen, Seitenangabe. Raum für eigene Einträge ist vorhanden.
- **ST_COMPX.PRG** zum Lesen, Suchen, Eingeben, Ausgeben. Es ist auf allen ST's in mittlerer und hoher Auflösung lauffähig.
- **ST_INDEX.EXE,** ein unter DOS lauffähiges Programm zum Lesen und Suchen der Dateieinträge.

ST_COMPX.PRG ist eine gekürzte Demoversion des Programmes ST_ARCHIVAR.

DM 29,–

ST BASIC-LIBS
NEU
DIE GEM-SCHNITTSTELLE FÜR GFA-BASIC 2.0

Mit diesem Programm erstellen Sie vollständige GEM-Programme, ähnlich wie mit **GFA 3.0,** jedoch ohne **Compiler-Probleme.**
- Mehr als 200 Einzeldateien
- Vollständige AES-Bibliothek
- Die wichtigsten VDI-Routinen
- Komplette GEMDOS, BIOS- und XBIOS-Routinen
- Einfachste Erstellung von Dialogboxen durch leichte Einbindung der Ressource-Dateien
- Viele Hilfsprogramme
- Umfangreiches Beispielprogramm
- Ausführliche Programmbeschreibung

DM 29,–

TKC-HAUSHALT ST
ORDNUNG, ÜBERBLICK, PLANUNG – MIT DER HAUSHALTSBUCHFÜHRUNG FÜR ALLE

Leistungsmerkmale
- Monats- und Jahrsabschluß
- Benennung von bis zu 50 laufenden und 200 Kostenkonten
- Verwaltung von bis zu 50 Dauerbuchungen mit wählbarer Frequenz
- Bis zu 2.500 Buchungen pro Periode
- Volle Eingabekontrolle auf Fehler
- Erstellung von Bilanzen, Journals, Saldenlisten und Kontenrahmen
- Ausgabe von Buchungslisten mit Korrekturmöglichkeit
- Saldenbildung von Kontenblöcken
- Anpassung an jeden Drucker
- Schnittstelle zu anderen Programmen
- Voll in GEM eingebunden
- UP-DATE-SERVICE
  Monatsabschluß mit automatischer Saldenübertragung in die Datei für den Folgemonat
  Ausdruck aller wichtigen Daten
  Einsetzbar auf allen ATARI ST und MEGA ST mit S/W und Farbmonitor
  Handbuch

DM 129,–

ST ARCHIVAR
DIE ÜBERZEUGENDE DATEIVERWALTUNG

- Unterstützt die Produktion wissenschaftlicher Texte
- Verwaltet und bearbeitet Zitate u. Literaturangaben und Ausgabe als Text
- Komfortable Nutzung für Video- u. Adressendateien
- Datenverwaltung mit Dateien von bis zu 19 Datenfeldern (38 in Vorbereitung und 200 Datensätzen
- Dynamisch erweiterbar
- Veränderung der Dateimasken
- Speicherung der Daten als DIF oder SDF Datei, damit Übernahme in Datenbank möglich
- Beliebige Programme nachladbar u. auszuführen ohne ARCHIVAR zu verlassen (z. B. 1st Word)
- mit allen Textprogrammen kompatibel, die ASCII einlesen
- Ausdruck von Karten (z. B. Bibliothekskarten)
- Listen-Ausdruck auf Knopfdruck
- Etiketten-Ausdruck
- Einfache Druckerprogrammierung
- Einfach bedienbar mit der Maus

ST-ARCHIVAR enthält zahlreiche Hilfsprogramme
- Bibliotheks-Suchfunktion (ganze Disketten durchsuchen lassen)
- Uhr-Einstellung
- Info-Datei über alle ARCHIVAR-Dateien erstellen
- Druckersteuer-Programm
- Kopierprogramm
- Sortierprogramm
- Erzeugt RAM-DISK G bis 1750 KB
- Automatisches Kopieren in die RAM DISK G beim Start

und das alles für **DM 89,–**

TKC-EINNAHME/ ÜBERSCHUSS ST
DIE KOMFORTABLE BUCHFÜHRUNG FÜR FREIBERUFLER UND KLEINGEWERBETREIBENDE.

Ohne Vorkenntnisse leicht zu bedienen.

Leistungsmerkmale
- Freie Wahl des Abschlußzeitraums (Periode)
- Voreinstellung von bis zu 5 Steuersätzen
- Einrichtung von bis zu 10 laufenden Konten
- Einrichtung von bis zu 200 Einnahme/ Ausgabe-Konten
- Erfassung von bis zu 2.500 Buchungen pro Periode
- Volle Eingabekontrolle auf Fehler (Löschen der letzten Buchung)
- Erstellung von Bilanzen
- Erstellung von Journalen
- Erstellung von Saldenlisten und Kontenrahmen
- Datenaufbereitung für USt.-Voranmeldung
- Ausdruck aller wichtigen Daten
- Anpassung an jeden Drucker
- Schnittstelle zu anderen Programmen
- Periodenabschluß mit automatischer Saldenübertragung in die Datei für die Folgeperiode
- Voll in GEM eingebunden
- Einsetzbar auf allen ATARI ST und MEGA ST mit S/W und Farbmonitor
- UP-DATE-SERVICE
  Ausgabe von Liquidität, Gewinn und Verlust
  Handbuch

DM 149,–

ST AKTIE
AKTIENVERWALTUNG

ST-AKTIE ist ein leistungsfähiges Aktienverwaltungs- und Chartanalyseprogramm für den Profi- und Hobbybörsianer
- GEM-orientierte benutzerfreundliche Programmierung
- profimäßige Darstellung von Charts a. stufenlose Kurz- und Langzeitbetrachtung
- beinhaltet Hilfsmittel zur Chartanalyse z.B.:
  - ★ frei wählbare gleitende Durchschnitte
  - ★ darstellbare Unterstützungslinien, Trendkanäle usw.
- Depot- und Erfolgsanalyse
  - ★ Erfolgsanalyse zur Entwicklung einzelner Titel
  - ★ Gesamtdepotanalyse mit zusätzlicher grafischer Darstellbarkeit
- Kursdarstellung und -abfrage in Tabellenform
- spezielle Dividendenanalyse (incl. steuerlicher Aspekte)
- eigene Kontoführung zwecks Gegenbuchung zum Wertpapiergeschäft oder einfach zur Führung des eigenen Girokontos (incl. Verwaltung der Umsätze einer Kreditkarte)
- große Kursbibliothek mit Kursen von über 70 deutschen und internationalen Standardtiteln mit aktuellen Kursen bis August 1988 (teilweise schon ab 1986 geführt) bereits auf Diskette vorhanden, die jedoch frei gestaltbar ist.

DM 79,–

Update **DM 19,–**

ST PLOT
KURVENDISKUSSIONS- U. FUNKTIONSPLOTTER-PRG.

- ist ein Programm, das mehr kann, als einfach nur Funktionen zeichnen. Es bestimmt die richtigen Ableitungen u. damit werden Null-, Extrem- und Wendestellen berechnet, es stellt den Definitionsbereich u. die Periode fest und vieles andere mehr.
- zeichnet die Schaubilder von bis zu 3 Funktionen u. deren erste u. zweite Ableitung gleichzeitig auf einen Bildschirm. Danach können Ausschnitte der Funktionen vergrößert und verkleinert werden.
- ist kpl. GEM- u. mausgesteuert. Die Bedienung ist somit sehr einfach. Sie brauchen nur die Funktion einzugeben, alle anderen Berechnungen erledigt das Programm.
- läuft sowohl in mittlerer wie auch in hoher Auflösung.
- ist nicht nur für Professoren, Studenten u. Schüler interessant. Es kann überall dort angewendet werden, wo man sich mit Funktionen beschäftigt.
- Umfangreiche Eingabemöglichkeiten, z. B. Verwendung sämtl. Funktionen eines wiss. Rechners. Defin. von 10 versch. Zahlenwerten als Konstanten, e u. Pi verwendbar.
- Komfortable Zeichnung, z. B. autom. Zeichnung u. Beschriftung der Achsen und deren Skalierung, Fehlerroutinen, damit keine unnötigen Asymptoten gezeichnet werden, beliebig viele Funktionen nacheinander in ein Bild einzeichenbar ermöglicht Funktionenvergleich u. Ablesung der Schnittpunkte etc.
- Beste Berechnung der Funktionswerte, z. B. Hohe Rechengeschwindigkeit, hohe Rechengenauigkeit.

DM 69,–

CHEMPLOT
Ein komfortables, leicht zu bedienendes Zeichenprogramm für chemische Moleküle

Hardwarevoraussetzungen: Läuft auf allen Atari ST mit wenigstens einem Megabyte Speicherraum und monochromem Monitor (SM124)
- voll GEM-unterstützte Bedienung
- Alle Zeichenfunktionen werden mit der Maus erledigt
- Insgesamt 48 Piktogramme (Icons) erlauben schnellen Aufruf fast aller Funktionen
- Viele Funktionen auch über Tastatur anwählbar
- Beliebiges Papierformat einstellbar
- Druckertreiber für 9-Nadeldrucker (Star NL-10, Epson-FX-kompatibel, bis 240★216 Punkte je Quadratzoll) und für 24-Nadeldrucker (NEC P6, bis 360★360 Punkte je Quadratzoll)

Grafikausgabe in drei Pixelformaten:
- Degas (32 034 Bytes)
- Screen (= Doodle) (32 000 Bytes)
- STAD (gepacktes Format)
- Moleküle können gedreht, gespiegelt, gestaucht, gedehnt, um 10 % vergrößert, um 10 % verkleinert und durch Verschieben einzelner Atome verzerrt werden
- Viele vorgefertigte Moleküle/Molekülfragmente:
- Cyclopropan bis Cyclohexadecan (16-Ring)
- Benzol, Cyclopentadien
- Steroidgerüst, Norbonan
- Cyclohexan-Sesselform
- Alkylketten
- Sulfon-, Carboxyl, Aldehydgruppe
  und außerdem können neu gezeichnete Moleküle auf Diskette abgelegt werden
- 6 Einfachbindungsarten: normal, gestrichelt (2 Arten), Keil (vor und zurück) und Treppe
- 2 Doppelbindungsarten mit jeweils drei Orientierungen
- 14 Pfeilarten
- Außerdem Rechtecke, Kreise und Ladungen
- Griechischer und lateinischer Plottzeichensatz
- Beliebige Höhe und Breite der Buchstaben
- Fettschrift, Unterstreichung, Kursivschrift, Hoch- und Tiefstellung: kein Problem!
- Kopier- und Verschiebefunktion
- Scherenschnitt- und Gummiband-Funktion erlauben die Manipulation ausgewählter Grafikbereiche
- Vielfältige Einstellungen möglich:
- Feste Bindungslängen
- Feste Winkel (Winkel kann angezeigt werden)
- Abstand der Bindung einer Doppelbindung
- Breite eines Keils bezogen auf seine Länge
- Höhe der Buchstaben auf Atompositionen bezogen auf die Bindungslängen
- Wirkungsradius der Schnappfunktion

DM 98,–

BESTELL-COUPON an Heim-Verlag, Heidelberger Landstraße 194, 6100 Darmstadt-Eberstadt

* alle Preise sind unverbindlich empfohlene Verkaufspreise

Bitte senden Sie mir: ____________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnr. ____________________

PLZ, Ort ____________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich



STARKE SOFTWARE

ST-KREATIV-DESIGNER
DAS SUPER-PROGRAMM ZUM KREATIVEN ERSTELLEN VON GRAFIKEN ALLER ART – OHNE MALKENNTNISSE

Das Selbstverständliche
Einfache Bedienung. Ansprechende Menüs. Alle Bildoperationen lassen sich rückgängig machen. Automatischer Malmodus, der Muster in Millionen Variationen erstellt.
- 2 Füllroutinen
- 2 Blockspeicher
- Absolut flimmerfreie Block- und Bildverarbeitung. Die eingestellte Verknüpfung ist schon beim Zeichnen und bewegen von Bildteilen zu sehen
- WYSIWYG! Blocks lassen sich in jeder Richtung über den Bildrand hinausschieben. Ausschneiden und Einkopieren von beliebigen Formen!
- Freihand: verschiedene Pinsel. Füllmuster als Pinsel, Malen mit Bildausschnitten. Spiegeln beim Zeichnen um 1 oder 2 Achsen
- Füllmuster definieren ohne Editor – eine ganz neue einfache aber wirkungsvolle Methode
- Vergrößern (Bild oder Text)
- Verkleinern, 3 Endprodukte zur Auswahl
- Spiegeln
- Drehen: dreht in 2 Richtungen gleichzeitig
- Stauchen
- Rahmen(1): Erzeugen von Rahmen aus allen Graphiken
- Rahmen(2): automatische Rahmenberechnung
- Schatten: automatische Schattenberechnung (3D Effekt)
- Teilmuster: aus Bildteilen (und Teilchen) können weitere Muster, auch Füllmuster erstellt werden
- Font's: 23 Größen, 21 Arten, 4 Verknüpfungen, weitere Verarbeitung durch Schatten. Rahmen usw. möglich. Schreibrichtung und Farbe kann während des Schreibens geändert werden.
- Lineal: Einblendbare Einteilung
- Radierer und Sprühdose in jeder Größe einstellbar. Zwei Sprühdosen Modi
- Schnelle Lupe mit: Punkt, Rechteck, Linie, Inventieren, Löschen
- Weitere Optionen: z. B. Bewegen, Kopieren, Kopieren nach..., Rechteck, Kreis, Linie, 3D-Rechteck, Preistafel...
- Invertieren auf Knopfdruck während gezeichnet wird
- Voll Mausgesteuert
- Kreativ-Designer – Bilder können in Signum 2 verarbeitet werden!
- Für Designer, Werbestudios, Druckgewerbe, Graphiker, aber auch für den nicht professionellen Anwender z. B. zum Erstellen einer Schülerzeitung, Handzettel, Grußkarten usw.

Das Besondere
- Fragen Sie mal die Bedienungsanleitung Ihres Zeichenprogramms ob folgende Funktionen möglich sind:
- Selbständige Hintergrund- und Füllmustererzeugung
- Über 200 Selbstdefinierbare Füllmuster im Speicher
- Kein Flimmern kein Gummiband
- Beim Zeichnen und Bewegen ist immer das fertige Endprodukt zu sehen
- Einige selbständige Rahmen und Schattenberechnungen
- Drei Vier Fünf... Neunundvierzigecke
- Automatisch Blocks ausschneiden
- Block paßgenau Einkopieren mit oder ohne Rahmen
- Blocks lassen sich in jede Richtung über den Bildrand hinausschieben
- Blaupause, Abmalen, Durchmalen von anderen Bildern
- Ein echtes Zeichenbrett mit Linealen
- Bilder maskieren
- Blocksatz, Proportionalsatz, Zentrieren

DM 128,–

* alle Preise sind unverbindlich empfohlene Verkaufspreise

SALIX DIALOGEXPERTE
Wer schon einmal eine Anwendung unter GEM geschrieben hat – mit Drop-down-Menüs, Dialogboxen und Alerts, wird es wissen:
So einfach sich die einzelnen Objekte mit dem Resource Construction Set Editor erzeugen lassen, so umständlich ist ihre Einbettung in das Anwendungsprogramm.
Es geht auch anders: SALIX hat jetzt einen DIALOG-Experten entwickelt, der die RSC- und die DEF-Dateien selbständig untersucht. Dann geht's erst richtig los: **„Programming by doing"** heißt die Device. Zunächst werden die Fenster der Anwendung definiert – mit der Muas. Dann wird die Bedeutung der Menüpunkte festgelegt, einfach indem man sie anklickt und das aufzurufende Prädikat eingibt.
Mehrere Menüleisten sind zulässig. Die einzelnen Menüpunkte lassen sich in logische Gruppen zusammenfassen, das Check-Symbol wird kontrolliert, und und und...
Die Knöpfe der Dialogboxen und Alerts lassen sich symbolisch ansprechen, eine Reihe von Prädikaten ermöglicht den einfachen Zugriff auf die GEM-Objekte von der Anwendung aus, komplexe Formulare können auch dynamisch aus der Anwendung heraus aufgebaut werden.
Sie haben die Wahl: sich weiterhin damit abzuquälen, Menüpunkte auszuzählen, oder sich lieber auf das Design und Logik Ihrer Anwendung zu konzentrieren.
Der Dialogexperte läuft unter SALIX PROLOG 2.6.

DM 79,–

ST-MAXIDAT
DIE DATEIVERWALTUNG FÜR DEN HOHEN ANSPRUCH
- Schnittstelle zu Textverarbeitungsprogrammen für Serienbriefe, Rechnungen, Mahnungen, ... (möglich mit allen ASCII-Editoren (1st Word, Tempus, ...); große Möglichkeiten, da z. B. Datumausgabe und Zugriff auf externe Textdatei)
- Schnittstelle zum Zeichenprogramm 'Kreativ-Designer' für Bilddatenverwaltung (Diashow für Werbezwecke, IC-Datenbank mit abgebildetem IC, Adressveraltung mit Foto/Stadtplan, bebildeter Anleitung) Formate: STAD/Neochrom/Degas
- frei erstellbare Druckerformate für beliebige Formulare, Etiketten, Listen, ...
- Rechnen im Datensatz (+ – ★/, Prozent) z. B. zur automatischen Ermittlung der MwSt sowie Bilanzvergleich (Gesamtergebnis, Durchschnitt)
- Druckeranpassung für alle Drucker im laufenden Programm erstellbar (Init, Exit, Drucktabelle für Umlaute, Druckersteuerung und sonstiges)
- mehrere Dateien verknüpfbar und daraus durch Selektierung neue erstellbar
- 'Filter' zur Beschränkung der Ausgabe von Datensätzen mit bestimmten Kriterien
- Automatisches Löschen von bestimmten Datensätzen (z. B. alle Kunden, die seit 1986 keine Bestellung mehr aufgegeben haben)
- Übersichtsdarstellung mit verschiedenen Zeichengrößen
- Suchen und Sortieren nach allen Datenfeldern sowie globales Suchen
- leistungsfähiger Editor zur Beschriftung der Datensätze
- Prg. arbeitet arbeitsspeicherorientiert, daher hohe Verarbeitungsgeschwindigkeit
- Programmaufruf aus dem laufenden Programm ohne Datenverlust
- konvertiertes Ausdrucken eines beliebigen Textes mit Zeilennr., Dateiname, Datum
- Uhrzeit und Datum neu stellbar
- Feldnamen jederzeit änderbar
- max. 100 000 Datensätze je Datei bei Mega ST-4
- voll GEM-gesteuert und einfachste Bedienung trotz vieler Funktionen

DM 98,–

SALIX-PROLOG
leistungsfähiges KI-Programmiersystem
- Edinburgh-Standard
- schnell, ca. 1200 LIpS
- ca. 160 eingebaute Funktionen
- Gleitkommaarithmetik
- läuft unter GEM
- ca. 140 GEM Library Funktionen
- Datenbankeditor
- Einbindung von Fremdeditoren
- zyklische Strukturen werden verarbeitet
- Exception Handling
- leistungsfähiges Testsystem

Jetzt neu: Version 2.6
- benutzerdefinierte Funktionen
- globale Variable
- Clipboard-Device

DM 198,–

Update von 2.0 – 2.5 DM 49,–

ST DISK BOX
DISKETTEN-VERWALTUNGSPROGRAMM
- Ablegen der Disketten nach Nummern (650 Stück)
- Ablegen in freibenennbaren Sparten
- Mitablegen von Texten möglich
- Suchen nach Programmname, Disknummer, Text, Extender
- Löschen durch Mausklick
- Druckfunktion für Disknummer, Extender, Sparte
- Programm erkennt doppelte Einsortierung in eine Datei
- Programm ist voll GEM-unterstützt
- große Geschwindigkeit
- komfortabel zu bedienen
- Nachträgliches Bearbeiten der Dateien möglich

DM 49,–

ST-C.A.R.
Computer Aided Regulation
Das Programm zur Systemanalyse und Regelungssynthese

Zielgruppe / Aufgabenbereich
Studenten der Ingenieurwissenschaftlichen Fachrichtung
- Veranschaulichung der Auswirkung von Regelkreisglieder/-strecken
- Überprüfung erarbeiteter Lösungen (numerisch und grafisch)
- Einsatz in der Entwicklung eigener Regelsysteme (z. B. Diplomarbeit)

Ingenieure der technischen Fachrichtung
- Einsatz in der Entwicklung benötigter Regelkreise (Zeitersparnis)
- Systemanalyse vorhandener Regelungen (Untersuchungen möglicher Änderungen)
- Unterstützung beim Entwurf durch 'weniger Erfahrene' (Auffrischung des Wissens, eigenes Wissen überprüfen)

Institute, Lehr- und Forschungseinrichtungen des technisch-wissenschaftlichen Bereiches
- Einsatz in der Entwicklung benötigter Regelsysteme
- Analyse vorhandener Regelsysteme (Untersuchung möglicher Änderungen)
- Hilfe bei der Erstellung von Lehr-(Unterrichts-)Material
- Direkter Einsatz im Unterricht ('am lebenden Objekt')

C.A.R. ...das Programm mit umfassender Leistung
C.A.R. ...ausführliche Anleitung mit Beispielen
C.A.R. ...das Programm für professionelle Ansprüche

DM 198,–

ST PRINT
Neue Version
DAS VIELFACH BEWÄHRTE MULTIACCESSORY JETZT IN ERWEITERTER VERSION V2.1 INCL. VIRUSKILLER

ST-PRINT erleichtert durch seine Vielseitigkeit und einfache Bedienung die tägliche Arbeit. Insbesondere Anpassungs- und Verständigungsprobleme zwischen Computer und Drucker werden gelöst.
Parameter -load und -save alle Einstellungen (Ramdisk, Spooler, Hardcopy und Druckervoreinstellung) werden abgespeichert und können auch beim Booten automatisch eingestellt werden.

1. Resetfeste Ramdisk
- Größe einstellbar von 32 – 4000 KB, auf Wunsch resetfest oder abschaltbar
- kann auf Laufwerk C bis P gelegt werden
- arbeitet problemlos mit einer Harddisk zusammen

2. Druckerspooler
- Größe einstellbare von 2 – 510 KB
- abschaltbar
- arbeitet mit TOS- und GEM-Programmen
- auch für Hardcopies
- in Maschinensprache programmiert und interruptgesteuert, dadurch immer optimale Geschwindigkeit beim Ausdruck
- Größe nun 2 – 4000 kByte

3. Druckervoreinstellung
- komfortable Druckereinstellung mit der Maus: Knopfdruck statt Handbuch oder DIP-Schalter
- viele Einstellmöglichkeiten: Zeilenvorschub, Schriftart, Zeichensatz, Papierränder etc.
- Einstellung des Druckers vom Desktop und aus jedem GEM-Programm (VIP-Professional, Wordplus, Tempus etc.) möglich
- **kann an alle Matrixdrucker angepaßt werden, diverse Druckertreiber gehören zum Lieferumfang, auch eigene Anpassungen können erstellt werden**
- **Druckeranpassung jederzeit nachladbar**

4. Druckerzeichenkonverter
- Anpassung aller Atari-Zeichen an den verwendeten Drucker, ermöglicht den korrekten Ausdruck von Umlauten und Sonderzeichen auf allen Matrixdruckern
- Umlaute und Sonderzeichen können gemischt verwendet werden
- endlich problemloser Ausdruck mit Programmen (TOS oder GEM), die keine eigene Druckeranpassung haben

5. Hardcopyroutine
- nutzt die Fähigkeiten von 9-, 18- und 24-Nadeldruckern
- kann an jeden Matrixdrucker angepaßt werden
- verschiedene Auflösungen, Schnelldruck bis Qualitätsdruck
- arbeitet mit dem eingebauten Spooler zusammen *gespoolte Hardcopy
- Auslösung der Hardcopy durch Alternate/Help, kann genauso wieder abgebrochen werden
- Umsetzung der Farben in Grauwerte (im MID-RES Modus)
- nun auch Ansteuerung HP-Laserjet-kompatibler Laserdrucker
- Hardcopy als Bild auf Diskette, Ramdisk oder Harddisk im Degas- oder Screenformat

6. Viruskiller
- Erkennen und Vernichten von BOOT-SEKTOR-Viren
- GEM-Programm, durch einfaches Umbenennen (.ACC statt .PRG) auch als Accessory verwendbar

DM 69,–

Update DM 19,–

BESTELL-COUPON
an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Bitte senden Sie mir: ______________________

____

zzgl. DM 5,- Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ______________________

Straße, Hausnr. ______________________

PLZ, Ort ______________________

Benutzen Sie auch die in ST COMPUTER vorhandene Bestellkarte.

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

SCHWEIZ
DataTrade AG
Langstr. 94
CH-8021 Zürich

Immer up to date

Mit dieser Sparte wollen wir allen unseren Lesern die Möglichkeit geben, sich über die neuesten Programm-Versionen zu informieren. Angegeben werden die aktuelle Versionsnummer, ein eventueller Kopierschutz, die Bildschirmauflösungen und der Speicherbedarf. Softwarefirmen ist es somit möglich, die ST-Computer-Leser über ihre Updates zu informieren.

Programmname	Version	Daten	Programmname	Version	Daten
Adimens ST	2.3	N HM	Interlink ST	1.89	N HM
Aditalk ST	2.3	N HM	K-Resource	1.1	N HM
Adress ST / Check ST	1.0	N H	Label ST	1.0	N HML
Afusoft Morse-Tutor	2.0	N HML	Laser C (Megamax)	1.2	N HML
Afusoft Radio-Writer	1.0	N HML	Laser Deluxe	1.0	N HML 1M
Afusoft Radiofax plus	1.0	N HML 1M	1st_Lektor	1.2	HM
AnsiTerm	1.4	N	Lern ST	1.22	N HML
1st BASIC Tool	1.1	N HML	Link_it GFA	1.1	N HML
Binlook	1.0	N HML	Link_it Omikron	2.0	N HML
BTX für ST	1.0	N H 1M	Lisp Complete	1.01	N HM
BTX/VTX-Manager	3.0	N H 1M	Lock_it	1.0	J HML
Calamus	1.01	N H 1M	Mega Paint	1.0	N H 1M
CIS-L&G	1.01		Megamax Modula 2	3.5	N HM
Creator	1.0	N H	Micro C-Shell	2.70	N HM
Crypt_it	1.0	J HML	MT C-Shell	1.20	N HM 1M
fibuMAN	3.0	N H	Multi ST	1.0	N HML 1M
fibuSTAT	2.3	N H	Musix32	1.01	J H
Flash-Cache/Flash-Bak	1.0	N HM	NeoDesk	2.0	N HML
Flexdisk	1.2	N HML	Omikron Assembler	1.0	N HML
1st_Freezer	1.0	N HML 1M	Omikron BASIC Compiler	3.01	N HML
GFA-Artist	1.0	N L	Omikron BASIC Interpreter	3.0	N HML
GFA-Assembler	1.2	N HML	PAM's TERM/4014	3.012	N H
GFA-BASIC 68881	1.3	N HML	PAM's TurboDisk	1.7	N HML
GFA-BASIC-Compiler	2.02	N HML	PAM's NET	1.0	N HML
GFA-BASIC-Interpreter	3.04	N HML	PC ditto Euroversion	3.96	N HML
GFA-Draft	2.1	N	Pro Sound Designer	1.2	L
GFA-Draft plus	3.0	N	Pro Sprite Designer	1.0	L
GFA-Farb-Konverter	1.2	N H	Search!	2.0	N HM
GFA-Monochrom-Konverter	1.2	N ML	Signum! zwei	1.0	N H
GFA-Objekt	1.2	N HM	1st_Speeder	1.01	N HML
GFA-Raytrace	1.5		1st_Speeder 2	1.0	N HML 1M
GFA-Starter	1.1	N HML	STAD	1.3	N H
GFA-Vektor	1.0	N	ST Pascal plus	2.02	N HM
Hard Disk Accelerator	1.0	N HML	SuperScore	1.3	J H 1M
Hard Disk Toolkit	1.05	N HM	Tempus	2.0	N HM
Harddisk Utility	2.0	N HM	Transfile ST plus	3.0	N HM
Imagic	1.1	N HML	VSH Manager	1.11	N HML 1M
Intelligent Spooler	1.01	N HML			

Irrtum vorbehalten!
Daten-Legende : N = kein Kopierschutz, J = Kopierschutz, H = hohe Auflösung, M = mittlere Auflösung, L = niedrige Auflösung, 1M = mindestens 1 Megabyte

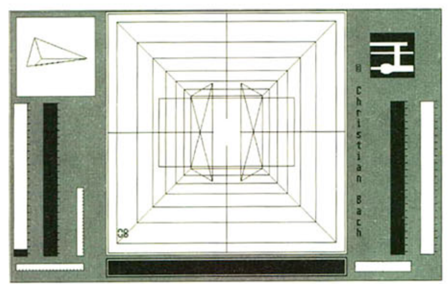
ST-COMPUTER PUBLIC DOMAIN

Das neue Jahr beginnt gut im Public Domain-Sektor. Wieder bereichern viele gute Programme unsere Sammlung. So z.B. Opus, ein 3D Vektorspiel, das mit schneller Grafik überzeugt. Auch im Bereich der Utilities gibt es diesmal wieder einige Leckerbissen: eine Fileselectorbox, ein Bildkonverter, eine Realtime-Bildschirmlupe und ein exzellentes XREF-Programm für BASIC.

Aber werfen Sie doch selbst ein Blick auf die neuen Disketten.

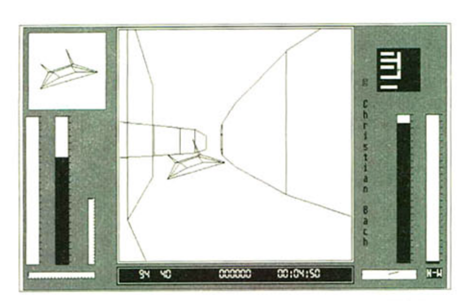


SPIEL



Das Tor zum Labyrinth öffnet sich

OPUS_1: 3D-Grafik Actionspiel. In räumlichen Gängen müssen Sie ein gegnerisches Flugobjekt finden und zerstören. Dabei steuern Sie Ihre Sonde mit dem Joystick durch das in Echtzeit dargestellte Labyrinth. Loopings und schnelle Wenden lassen sich ebenso leicht fliegen wie rasante Kurven. Doch Vorsicht, schon im nächsten Gang lauert der Gegner. (s/w)

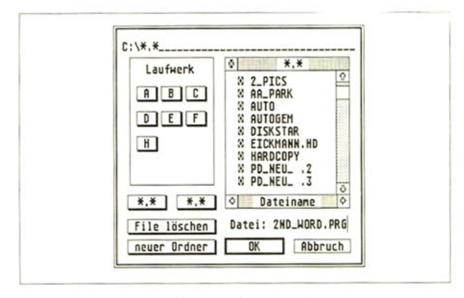


Plötzlich taucht aus dem Nichts, ein feindliches Objekt auf. Angriff oder Flucht, das ist hier die Frage?



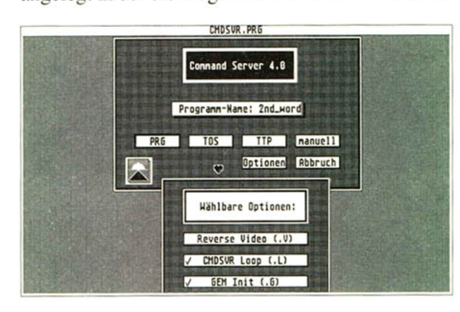
UTILITIES

FILEBOX: Fileselectorbox, die sich automatisch einklinkt und daher die Original-Fileselectorbox ersetzt. Belegt nur 7KByte Speicher. Laufwerksymbole direkt zum Anklicken. Anzeige der Programmlänge, Zeit und Datum. Bis zu 16 Laufwerke.

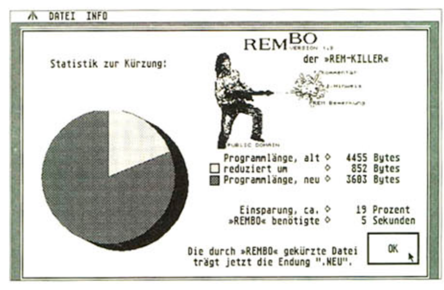


Die Fileselectorbox bietet viele Vorzüge gegenüber dem Original.

CMDSVR: Ermöglicht das Starten eines Programmes, von dem Sie nicht wissen, wo es gespeichert ist. Dabei wird einmalig eine Suchpfadliste angelegt in der die möglichen Pfade enthalten sind.

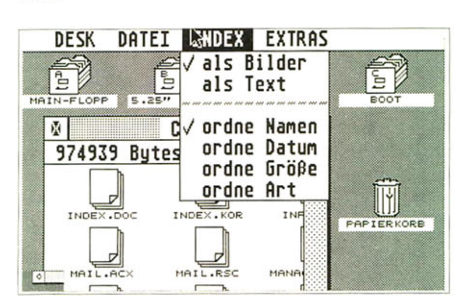


REMBO: Entfernt alle REMS (Bemerkungen) aus einem GFA-BASIC-Programm. Ein Programm wird dadurch wesentlich kürzer. (s/w)



REMBO eliminiert auch den kleinsten Kommentar

GOODVIEW: Diese Lupe zeigt eine Ausschnittsvergrößerung des Bildschirm. 2-,4-,8- und 16-fache Vergrößerung. Aktivierung mit 'Alt-Shift-Shift'. Läuft mit GEM- und TOS-Programmen.



Schon die 2fache Lupe, läßt einige Feinheiten erkennen.

CONVERT: Konvertierprogramm zwischen verschiedenen Grafikformaten (DEGAS, STAD, PWORKS, DOODLE, IMG, ART DIRECTOR, NEOCHROME) und verschiedenen Auflösungen (niedrig, mittel, hoch). Dabei werden auch die gepackten Formate berücksichtigt.

WT-HD: Dateisuchprogramm für Festplatte. Geschwindigkeitsoptimiert. Suche mit Wildcards (z.B: *.DOC). (s/w)

SUCHSETS: Zeigt die in einem Signum-Dokument verwendeten Zeichensätze. Sinn ist das gezielte Kopieren dieser Zeichensätze auf eine Diskette oder die Ramdisk. (s/w)



UTILITY

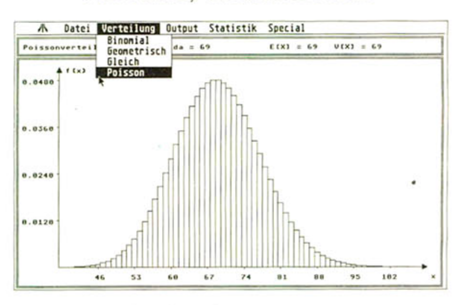


Vor unsauberer Programmierung wird gewarnt

WELLER-TOOLS: Cross-Referenz-Analyser für globale und lokale Variablen sowie Übergabeparameter, Labels und Prozeduren; findet Fehler und macht Verbesserungsvorschläge; Ausgabe als Zeilen- und Prozeduren-Referenz. Outlinefunktion für Analyse der Programmstruktur. Zahlreiche Hilfsroutinen. Für GFA-BASIC 2.0, 3.0 und OMIKRON.BASIC.



MATHEMATIK, STATISTIK, PHYSIK, GEOMETRIE

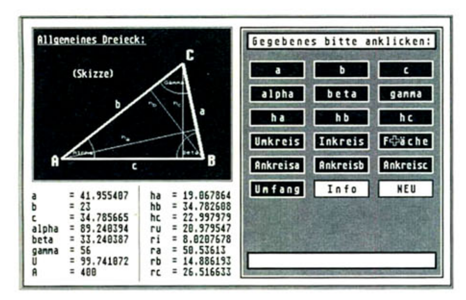


Eine Poisson-Verteilungskurve

DISKRET: Grafische Darstellung diskreter Verteilungsfunktionen; binominal, geometrisch oder Poisson. Die Ausgabe ist zur weiteren Verarbeitung zusätzlich auf Datei möglich. (s/w)

DREIECK: Das Programm berechnet aus drei typischen Wertangaben eines Dreiecks sämtliche andere. (s/w)

ST-COMPUTER PUBLIC DOMAIN



DREIECK spart lästiges Kopfrechnen

PENDEL: Simulation eines Drehpendels mit variabler Dämpfung und Auslenkung. (s/w)

KÖRPER88: Berechnung geometrischer Körper. Würfel, Prisma, (Hohl)-zylinder, Pyramide, Kegel, Pyramidenstumpf, Kugel, Kugelabschnitt, -zone, Rechteck, Raute, Parallelogramm, Trapez, Dreieck, Kreis, Kreisausschnitt, -abschnitt, -ring, Ellipse. (s/w)

KREDIT: Rechnungen aus dem Kreditwesen. Kosten, Effektivzins, Vergleichszins, Unterschiedsbetrag, Rediskont. (s/w)



GESCHICHTE

Geburtstag zb.:03.08 Name - erster Buchstabe groß zb.Friederich

Bedeutende Geschichtliche Ereignisse

Geben Sie einen Männlichen Vornamen ein ? Andreas

Andreas Schlüter 1660 - 1740 dt. Baumeister. Ab 1694 am Berliner Hof.

Andreas Hofer 1767 - 1810 Südtiroler Freiheitsheld

Andreas Celsius 1701 - 1744 schwed. Astronom. Erfand das Thermometer. Trägt seinen Namen

Andreas bedeutet der Mannhafte

Soll ich noch einen Vornamen suchen j/n

Auf den Spuren historischer Berühmtheiten und Ereignisse

HISTORY: Nennt zu jedem weiblichen/männlichen Vornamen bekannte historische und neuzeitliche Personen des gleichen Namens. Ebenso sind mehr als tausend geschichtliche Ereignisse und Geburtstage bekannter Persönlichkeiten gespeichert. Interessantes Nach'tipp'Werk. (s/w)

PD 188

QUIZ

The Braingame ST

BRAIN für alle Freunde von Quizspielen

BRAIN: Quizspiel für Anspruchsvolle. Genau 1001 Fragen aus verschiedenen Wissensgebieten werden gestellt. Das Niveau ist recht hoch. Für Freunde von Trivial Pursuit sehr zu empfehlen. (s/w)

Wir verwenden ausschließlich doppelseitige Markendisketten (2DD) der Firmen Maxell und JVC. Sämtliche Disketten sind geprüft, fehlerfrei und virenlos.

UPDATES

Folgende Disketten wurden aktualisiert:

175 SAKROTAN V 4.05: Nun auch mit kompletter Überprüfung einer Diskette oder der gesamten Festplatte auf Linkvirus. Weitere Vergleichsviren in der Bibliothek.

142 SCANNER 1.4: Software arbeitet nun auch in mittlerer Auflösung. Herstellung von Bildausschnitten. Schnellere Berechnung.

Die neuen Versionen sind ab dem Erstverkaufstag dieser ST-Computer auf den entsprechenden Disketten enthalten.

Es gelten die normalen Versandbedingungen.



Auf der nächsten Seite finden Sie weitere Disketten aus unserer Sammlung



PD-NEWS 2 ist erhältlich bei:

MAXON Computer GmbH
'PD-Service'
Industriestr. 26
6236 Eschborn

In PD-NEWS 2 lesen Sie unter anderem folgende Artikel:

- **Die Hyperstory:** Wissenswertes über Hyperformat und dessen Autor Claus Brod
- **SAKROTAN:** Computerviren und deren Beseitigung
- **Hochzeit:** Ein Beispiel linearer Optimierung mit ORSPREAD
- **KI zum Nulltarif:** XLISP 2.0 und deren Folgen
- **FSELECT:** Geheimnisse des neuen Fileselect-Standards

Programmierpraxis: Hardcopy ohne Streifen (GFA-BASIC)

Hardwarebastelei: JOYSTICK mit Dauerfeuer - Schonung des Daumens

Unterhaltsames: Computer und die Nachbarn - ein Bericht aus den täglichen Leben eines Computerfreaks

Auch als **Nachschlagewerk**:
Enthalten ist eine komplette, ausführliche Liste unserer PD-Sammlung. Hier finden Sie jedes Programm mit Kurzbeschreibung, von 1-175.

Preis: DM 2.50 (+ DM 1.50 Porto)
Bei zusätzlicher Bestellung von PD-Disketten enfallen die DM 1.50 Porto.

ST-COMPUTER PUBLIC DOMAIN

Altbewährtes

Die nicht aufgeführten Disketten sind natürlich auch weiterhin erhältlich, nur wurde uns der Platz zu klein. Schauen sie dazu bitte in einer der vorherigen Ausgaben nach oder bestellen Sie die PD-NEWS.

Diskette 172
OMIKRON UPDATE 3.0 -> 3.01
Allen glücklichen Besitzern des neuen OMIKRON.BASICs bieten wir im Zusammenarbeit mit der Firma OMIKRON. einen einmaligen Update-Service. Das Programm auf dieser Diskette dated Ihren OMIKRON.-Interpreter V3.0 auf die neueste Version 3.01 up.
Neben kleinen Änderungen am Interpreter, generiert dieses Programm eine überarbeitete GEMLIB (belegt nur noch eine Programmzeile) mit unverzichtbaren Funktionen, und alle anderen Dateien, die zur Original-OMIKRON-Diskette gehören (siehe NEWS).
Achtung: Das Update-Progamm prüft nach, ob Sie Version 3.0 schon besitzen und generiert nur dann die neue Version 3.01.

Diskette 173
3D UND SIMULATIONEN
-ERDKUGEL: Realtime-Animations-Programm. Die Erdkugel wird mit 4 Bildern pro Sekunde im Raum bewegt. Dabei können Drehung, Lage und Entfernung per Tastatur manipuliert werden. "Scotty beamen".
-LIFE 3D: Eine räumliche Variante des Generationenspiels. Durch die dritte Generation ergeben sich neue Aspekte. (s/w)
-EDIT 3D: Programm zur räumlichen Darstellung von Körpern. Diese können frei definiert und im Raum bewegt werden. (s/w)
-PENDEL: Simulation eines Drehpendels. Der Bewegungsablauf wird mit Hilfe von DGLs berechnet und grafisch dargestellt. Interessantes Programm für's Physik-Labor zu Hause. (s/w)
-FILTER: Programm zur Berechnung verschiedenster Filter: IIR- und FIR-Filtertypen, Bessel-, Butterworth- und Tschebyscheffilter mit verschiedenen Welligkeiten. Dabei werden Band- Hoch-, Tief- und Allpaß berechnet. Neben den Werten wird der Signalpegel grafisch dargestellt. (s/w)

Diskette 174
SPIELE
-TUZZLE: Originelles Computerspiel. Durch Verschieben von Bausteinen (Rohrleitungen) muß dem einfließenden Wasser ermöglicht werden, von links nach rechts zu fließen. Dabei ist Überblick und schnelle Reaktionszeit gefragt, denn sonst ist das Wasser schneller. Ein unterhaltsames Spiel für Joystick und s/w-Monitor. (s/w)
-PUSH_ME: Spiel für scharfe Denker. Auf einem Spielfeld muß ein Viereck von der Start- auf die Zielposition geschoben werden. Das hört sich aber leichter an, als es ist. In der Redaktion waren jedenfalls einige Personen, viele Stunden lang, anstatt ihrer Arbeit nachzugehen, am Steine-Schieben.
(s/w)
-BUMERANG: Arkanoid-Variante mit Bumerang statt Ball. Dadurch ergibt sich ein besonderer Bewegungseffekt, denn durch Anschneiden des Bumerangs fliegt dieser eine Kurve. (s/w)
-LANDER: Notlandung eines Raumschiffs auf einem Planeten. Der Blick aus dem Cockpit zeigt die Entfernung und den Landezielpunkt. Durch geschickte Richtungs- und Schub-Steuerung müssen Sie nun versuchen das Raumschiff sicher zu landen. Wenn es nicht funktionieren sollte (passiert recht oft), gibt das Programm eine Auswertung des Absturzes. (s/w)

Diskette 175
ANTIVIRUS & UTILITIES
-SAKROTAN 4.05: VIREN auf Ihrem ST? Keine Panik. SAKROTAN erkennt sie und desinfiziert die betroffenen Disketten. SAKROTAN unterscheidet durch seine Bibliothek zwischen bösartigen VIREN und anderen Bootprogrammen (TOS-Lader, Aladin-Booter, 60Hz-Umschalter, u.a.). Weiterhin erkennt es LINK-VIREN, indem es das betreffende Programm auf kritische Sprünge analysiert.
Ein spezielles Boot-Programm schützt die Disketten vor erneuten VIREN-Befall. Kurz ausgedrückt: SAKROTAN schützt vor allen bislang bekannten VIREN. (s/w)
Ausführlicher Bericht zu Sakrotan in der neuen Ausgabe der PD-NEWS. Sofort mitbestellen - es lohnt sich!

Machen Sie mit!

Möchten Sie ein selbstgeschriebenes Programm in die PD-Sammlung geben? Kein Problem. Schicken Sie es uns auf einer Diskette zu, samt einer Bestätigung, daß es von Ihnen erstellt wurde und frei von Rechten Dritter ist.

-BCS: Das BASIC Construction Set unterstützt die Programmierung von eigenen Desktops mit GFA-BASIC. Durch ein eingebautes Malprogramm können die Menüs (Buttons und Verzierungen) auf dem Bildschirm gezeichnet und Icons plaziert werden. BCS erzeugt dann ein ablauffähiges GFA-BASIC-Programm, das diesen Desktop aufbaut und die eingebauten Buttons abfragt. (s/w)
-G_C SHELL: Shell für GFA-BASIC. Mit diesem Programm können Interpreter und Compiler bequem aufgerufen werden. (s/w)

Diskette 176
DRUCKUTLITIES
-BIG BANNER: Programm zum Drucken von Endlos-Bannern. Erzeugt meterlange Schriftzüge in verschiedenen Schriftarten und -größen. (s/w)
-LCV10 COLOR: 1st_ Word-Druckertreiber für STAR LC10 Color. Ermöglicht vielfarbigen Textausdruck. Auch für andere Farbdrucker anpassbar.
-P2000_INST: Einstellungsprogramm für sämtliche Steuercodes per Accessory. Zeilenabstand, Schrifttyp, Formatierung, Seitengröße,...
-NL10_INST: Weiteres Einstellprogramm für NL10. Als Accseeory kann der Drucker jederzeit nach eigenen Wünschen konfiguriert werden. (s/w)

Diskette 177
ELEKTRONIK
-TRANSISTOR: Entwicklung von elektronischen Schasltungen. Ausgeklügeltes Zeichenprogramm mit umfangreicher Symboleditor (Spiegeln, Verschieben, Drehen). Dadurch lassen sich auch ausgefallene Bauteile generieren. Blockoperationen, Lupe und sonstige Standardfunktionen. Die abgebildete Schaltung läßt sich in kurzer Zeit anfertigen. (s/w)

DISKETTE 178
ADVENTURE
-THE VAULT: In einem alten schottischen Schloß wird eines Morgens der Schloßherr an einem Baum erhängt aufgefunden. Die Indizien deuten auf einen Mord, doch die Polizei tappt im Dunkeln. Einige Hinweise sind gegeben, die restlichen müssen Sie im Schloß finden.
Es empfiehlt sich das Kartographieren der vielen Gänge, denn sonst verirrt man sich und wird den Fall nie lösen.
Das Adventure ist komplett in deutscher Sprache geschrieben und soll Ihnen helfen, die Winternächte zu überstehen.
Die Redaktion nimmt Lösungen des Spiels gerne entgegen.

Diskette 179
SPIEL
-EUROPA: Actionspiel mit Joysticksteuerung. Eine finstere Gestalt hat in Europa mehrere Bomben verteilt, die Sie als tapferer Agent finden, entschärfen und vernichten müssen. Dazu sind vier Level zu bewältigen, bei denen viel Fingerspitzengefühl gefragt ist. (s/w)

Diskette 180
WISSENSCHAFT
-EPHEMERIDEN: Das Programm dient dazu, nach Eingabe eines bestimmten Zeitraumes innerhalb eines Jahres die Ephemeriden verschiedener Objekte zu berechnen. Zusätzlich wird die Möglichkeit geboten, die Mond- und Sonnenfinsternisse eines Jahres zu berechnen, sowie die Konstellation der vier inneren Jupitermonde auf den Bildschirm zu plotten. Programm mit langer Sternwartenerfahrung. (s/w)

Diskette 181
SCHULE
-ARTUS: Rechentrainer für die unteren Schulklassen. ARTUS, der Rechentrainer, führt mit netten Auflockerungen an das leidliche Thema Mathematik heran. (s/w)
-KOPFRECHNEN: Rechentrainer für die Rechenarten +,-,*,/ und Wurzel. Verrschiedene Schwierigkeitsstufen von leicht bis reichlich schwer. Als Gegenpol für Taschenrechnerverwöhnte Hirne. Nicht nur für Kinder. (s/w)
-RECHEN: Wirtschaftsrechnen für Bürokaufleute. Umrechnen von Maßeinheiten, Mathematik, Dreisatz, Prozentrechnen, u.v.a.m.
-FASTTEST: Reaktionstrainer. Einfaches Programm zum Überprüfen Ihres Reaktionsvermögens. (s/w)
-STUNDENPLAN: Stundenplan-Designer für Schüler und Lehrer (mehrere Klassen). (s/w) (mit GFA 2.0-Source)

Diskette 182
-STECKWEG: Der freundliche Helfer zur Beseitigung überflüssiger Pfunde. Generiert Abnahmeplan und informiert über den Kaloriengehalt vieler Nahrungsmittel. Druck einer Kalorientabelle. (s/w)
-ALKOHOLTEST: Informatives Programm über Alkoholwerte.
Unter Berücksichtigung persönlicher Körpermaße berechnet das Programm den theoretischen Blutalkoholwert, die Abnahmekurve und informiert über Wirkung und Gefahren. Weiterhin wird die Anfälligkeit und vielzählige Fragen geurteilt. (s/w)

Achtung beide Programme ersetzen keine Fachberatung!

Zeichenerklärung:
s/w = nur monochrom
f = nur Farbe
keine Angaben = s/w und f



Sonderdisks

Die folgenden Programme sind nicht Public-Domain.
Sie können aber bei uns bezogen werden.

***A.)* TOS**: Das Betriebssystem auf Diskette vom 6.2.1986. Was tun, wenn Programme mit Blitter-TOS nicht laufen? Einfach das alte TOS von Disk laden.
Unkostenbeitrag DM 15.-

***B.)* RCS**: Das Resource-Construction Set aus dem ATARI-Entwicklungspakets. Unverzichtbar bei der GEM-Programmierung.
Unkostenbeitrag DM 15.

***C.)* EXTENDED VT 52 Emulator**

Schneller als der Blitter
Erweiterte Version des in der ST-Computer ausführlich vorgestellten Emulators. Das Programm ersetzt den im TOS integrierten VT 52-Emulator vollkommen. Enthält neue Routinen zur Bildschirmausgabe, die wesentlich schneller sind, als die im TOS eingebauten. Daher erfolgt eine Beschleunigung der Textausgabe um den Faktor 3-5.
Programmierung
Neben der Beschleunigung enthält der Emulator zusätzliche, programmierbare ESC-Funktionen, wie z.B. das Softscrolling (1 Pixel) nach allen vier Seiten. Wurde es eingeschaltet, so scrollt eine Zeile oder ein Textblock ohne Rucken. Ferner können die Textattribute (fett, unterstrichen, hell und kursiv) ohne GEM umgeschaltet werden. Auch die Übertragung von Grafik via VT 52 ist möglich.
Der Extended VT 52 Emulator wurde in vier Teilen ausführlich in der ST-Computer vorgestellt.
Diese Diskette ist nicht public-domain, kann aber über diesen Service von uns bezogen werden.
Unkostenbeitrag DM 15.-

BITTE BEACHTEN

Sämtliche Disketten können ab dem Erstverkaufstag der ST-Computer direkt bei der MAXON-Computer bezogen werden. Wir haben für Sie den schnellstmöglichen Versandservice eingerichtet. Lieferung innerhalb einer Woche.
Bitte beachten Sie folgende Punkte:

1. Schriftliche Bestellung

- Der Unkostenbeitrag für eine Diskette beträgt DM 10,-
- Hinzu kommen Versandkosten von DM 5,- (Ausland DM 10.-)
- Bezahlung nur per Scheck oder Nachnahme (Im Ausland nur Vorrauskasse möglich)
- **Ab 5 Disketten entfallen die Versandkosten (DM 5.- bzw. DM 10.-)**
- Bei Nachnahme zuzüglich DM 3,70 Nachnahmegebühr
- Bitte fügen Sie keine anderen Bestellungen oder Anfragen bei.

Adresse:

MAXON-Computer GmbH
'PD ST-Computer'
Postfach 5969
D-6236 Eschborn

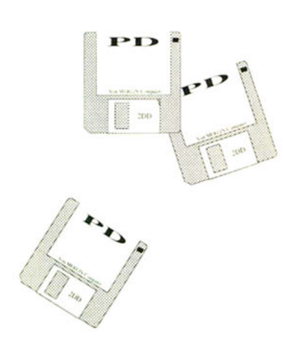
2. Anruf genügt

MAXON-Computer GmbH

'PD-Versand'
Tel.: 0 61 96 / 48 18 11
Mo-Fr 9⁰⁰ - 13⁰⁰ und 14⁰⁰ -17⁰⁰ Uhr

- Nur gegen Nachnahme (Gebühr DM 3,70)

Bei Fragen bezüglich der Programme stehen wir Ihnen gerne zur Verfügung.



INSERENTEN-VERZEICHNIS

Weide Elektronik GmbH
Regerstraße 34
D-4010 Hilden
Telefon (0 21 03) 4 12 26

Ladenlokal: Gustav-Mahler-Straße im Einkaufszentrum
Telefon Ladenlokal (0 21 03) 3 18 80 · Telefax (0 21 03) 3 18 20
viele Parkplätze direkt am Laden

ATARI System-fachhändler

Leasing für Komplettsysteme möglich
Atari + PC XT/AT

Schweiz
Computer Trend AG
Langstrasse 31
CH-8021 Zürich
Tel. (01) 2 41 73 73

Niederland
COMMEDIA
Leidsekade 98
1017 PP Amsterdam
Tel. (0 20) 23 17 40

SOFTWARE

SOFT

SINUS 79.-

Software für Schule und Labor
Grafische Darstellung von bis zu 8 Funktionen, deren Ableitungen und/oder Meßwertreihen, Kurvendiskussion, symbolische Ableitungen von Funktionen, numerische Integration, Schnittpunkt zweier beliebiger Funktionen, Interpolation mit kubischen Splines, lineare und Polynomregression

UNISHELL 98.-

Neues selbstgestaltetes Desktop
Interessantes Utility für alle Festplattenbesitzer. Alle Programme im direkten Zugriff. Keine lange Suche nach Ihren Programmen, direkt auswählbar in selbstgestalteten Drop Down Menüs. Teilbares Menü bei zu vielen Einträgen. Einfachste Installation durch Auswahl mit der Maus. Parameterübergabe an Programme (z. B. Dokumentübergabe an Wordplus), Schleifenprogrammierung (GFA-Basic mit Fileübergabe, Compilierung, Programmtest und wieder in den Interpreter), kopieren, löschen und vieles mehr!

CRYPT 79.-

Professionelles Ver- und Entschlüsselungsutility. Schützt Ihre Dateien vor unbefugter Benutzung! Verdeckte Schlüsseleingabe, extrem schnell (ver-/entschlüsselt 2,5 MB in ca. 70 Sek. auf der Festplatte), INM.ACE Verschlüsselungen, dynamisches Verschlüsselungsverfahren – äußerst sicher! Als Programm oder Accessory nutzbar.

68881 + WCL

(Weide Coprozessor Language)

Ein starkes Gespann

Mit WCL erhalten Sie mit dem 68881 auf 68000er Systemen die höchstmögliche Rechengeschwindigkeit. Denn WCL erzeugt linearen Assemblercode unter Ausnutzung aller koprozesserinternen Features wie Register und Konstanten! Und dabei können Sie Ihre Funktionen und Formeln äußerst einfach in pascalähnlicher Syntax programmieren. Sie ersparen sich eine Menge Zeit! Fordern Sie unser INFO an! Demodiskette + Handbuch für DM 30.-

Textverarbeitung + DTP

Tempus 2.0	119.-
1st Word plus	189.-
1st proportional	89.-
Protext ST	139.-
Signum 2	398.-
Signum Fontdisketten:	
Professional Fontdisk	100.-
Fontdisk Rockwell	100.-
Fontdisk Julia	100.-
Eurofont Diskette	100.-
Signum Utility	89.-
Calamus jun.	348.-
Calamus prof.	998.-

Tabellenkalkulation

LOGISTIX	398.-
VIP professional	289.-
BASiCALC	78.-
Querdruck f. Tabellenkalkulation	58.-

Bücher

ATARI ST Profibuch Sybex, 988 S.	69.-
Motorola 68000	59.-
Motorola 68020	79.-
Motorola 68030	98.-
Motorola 68881	79.-

Programmiersprachen

Omicron Assembler	99.-
GFA Assembler	149.-
Turbo C Assembler/Debugger	189.-
Omicron Basic Compiler	169.-
Omicron Basic 68881 Compiler	219.-
Ornikron Basic Libraries (Midi, Statistik, Numerik, Complex, Easygem)	je 79.-
(Finanzmathem., inlineassembler)	je 129.-
Maskeneditor, Link it, Draw	je 79.-
GFA Basic Interpreter 3.0	189.-
GFA Basic 68881 Int. + Comp.	298.-
Aztek C prof.	398.-
Aztek C developer	598.-
Aztek C S. L. Debugger	148.-
Laser C	388.-
Lattice C	248.-
Mark Williams C	298.-
Prospero C	248.-
Turbo C	189.-
Turbo C + Assembler/Debugger	279.-
AC Fortran	448.-
Prospero Fortran	348.-
SPC Modula 2	348.-
Pro PASCAL	248.-
OS-9	1598.-

Utilities

Harddisk Utilities (Applic. Syst.)	69.-
Crunch (Opt. Harddiskzugriff, Backup)	69.-
Anti Viren Kit	78.-
UNISHELL	98.-
NEO Desk	89.-

Emulatoren

PC DITTO V3.64	178.-
Aladin V3.0 mit ROMs (unterst. Harddisk + ATARI Laserdr.)	598.-

Spiele

Psion Chess	69.-
Flight II	99.-
European Scenery Disk	59.-
Jet (F16/F18 Flugsim.)	99.-

Kaufm. Software

BS Handel	498.-
BS Fibu	548.-
fibuMAN	798.-

CAD + Grafik

Campus Prof.	798.-
Campus Art	149.-
Campus Draft	149.-
CAD Project	298.-
CAD 3D Cyberstudio	178.-
GFA Draft PLUS	329.-
STAD	169.-
Imagic	478.-
Spektrum 512	148.-

HARDWARE

PROFESSIONAL SCANNER CANON IX12F 3998.-

Technische Daten:

Scannerelement:	CCD Zeilensensor
Scannertyp:	Flachbett mit innenlaufendem Schlitten
Belichtungslampe:	Halogenlampe 50 V / 42.5 W
Zulässige Originale:	Bücher, Blätter, Objekte
Abtastfläche:	216 × 297 mm (DIN A4)
Auflösung:	300 × 300 Punkte/Zoll (300 DPI)
Scanoptionen:	300/200/150/75 DPI
Grauwerte:	32 Graustufen, softwaremäßig erhöhbar
Dichteeinstufung:	3 Stufen
Moduswahl:	binär/dithered
Scangeschwindigkeit:	<16 Sekunden pro DIN A4 Seite
Abmessungen:	368 (B), 570 (T), 83 (H) [mm]
Gewicht:	114 kg
Spannungsversorgung:	100 – 115 V / 200 – 240 V
Bildformate:	MG, Postscript, Degas
Software:	Scansoftware, Retuscheprogramm

SPEICHERKARTEN steckbar

auf 1 MB für 260/520 STM	a. A.
auf 2 MB für 520 ST+/1040	a. A.
auf 2,5/4 MB	a. A.

Jede Erweiterung einzeln im Rechner getestet! Sehr einfacher Einbau ohne Löten. Gut bebilderte Einbauanleitung. Vergoldete Mikrosteckkontakte, dadurch optimale Schonung des MMU Sockels. Kein Bildschirmflimmern, keine zus. Software, keine zus. Stromversorgung!

ECHTZEITUHR PLUS 129.-

Keine Software nötig bei Betrieb mit Blitter TOS. Jede Uhr im Rechner getestet und gestellt. Interner Einbau ohne Löten. Dadurch freier ROM Port. Hohe Genauigkeit. Schaltjahrerkennung, Datum, Uhrzeit.

ICD HARDDISKS

herausgeführter SCSI Port zum Anschluß von Standard PC Peripherie, unterstützt bis zu 8 SCSI Geräte
- DMA Eingang, DMA Ausgang
- 100 % kompatibel zu ATARI Harddisks
- sehr leister Lüfter
- alle Harddisks laufen auch mit TURBO DOS
- eingebaute ECHTZEIT-UHR
- Treiber mit oder ohne Verify
- extrem leistungsstarke Software

85 MB form.	2498.-
Harddisks von 20 – 160 MB	a. A.

Einbau von 50 MB Harddisks in ATARI Mega ST preisgünstig und schnell

STREAMER a. A.

Übertragungsrate **5 – 6 MB/Minute!**
Komfortable Software!
Subsysteme aus Harddisk + Streamer a. A.

ST HOST ADAPTER

Anschluß von IBM komp. Harddisks an Ihren ATARI ST
- unterstützt bis zu 8 SCSI Geräte
- DMA Eingang, DMA Ausgang, SCSI Port
- integr. ECHTZEIT-UHR
- Softwareunterstützung für über 100 versch. Harddisks und Controller
- TOP Software

ST HOST ADAPTER	348.-
ADAPTEC 4000A/Omti 3520 MFM Controller	298.-
ADAPTEC 4070/Omti 3527 RLL Controller	398.-

FLOPPYLAUFWERKE

3.5" Einzellaufwerk	289.-
5 1/4" Einzellaufwerk	398.-

COPROCESSOR 68881 ab 498.-

Für 520/1040/Mega ST **inkl. Softwarelibraries** für Megamax C, Mark Williams C, DRI C, Lattice C, Pro-Fortran 77. AC Fortran, Modula II, CCD Pascal. Einfacher Einbau ohne Löten! INFO anfordern!

VIDEOSOUND BOX 248.-

Ihr ST am Fernseher mit durchschlagendem Sound! Klangkräftige 3 Wege-Box mit integriertem HF Modulator zum Anschluß aller ATARI ST an den Fernseher.

BLITTER Nachrüstplatine	79.-
BLITTER	198.-
TURBO-DIZER	348.-
JUNIOR PROMMER	189.-
EPROMKARTE 64 kB	12.90
Trakball	98.-
Ersatzteile	a. A.
Floppystecker, Floppybuchsen, Monitorstecker, Monitorbuchsen	8.90
Floppykabel	19.90
MMU/GLUE/BLITTERSOCKEL	29.90

COMPUTER-PROJEKTIONS-PANEL

Für alle die etwas präsentieren müssen!

Gestochen scharf, 8 Graustufen, hochmodernes DST LCD, mit IR Fernbedienung!
Geeignet für jeden Overhead Projektor.
Invertier Funktion, Löschfunktion, auch an IBM PS/2, IBM PC/XT/AT mit CGA, EGA, VGA Karte anschließbar!
Keine Spezialsoftware nötig
Auflösung 640★480 Bildpunkte, nur 3 kg Gewicht!

Komplettes **DTP System** bestehend aus Computer, Festplatte, Streamer, Monitor (oder Ganzseitenbildschirm) Software und evtl. Scanner a. A.
PRIVATLIQUIDATION für Ärzte incl. Unfallabrechnung a. A. **Komplette EDV Auftragsabwicklung** a. A. **Komplette EDV Buchhaltung** a. A. Vorführung nach Absprache! Sonderkonditionen für Wiederverkäufer

In der nächsten ST-Computer lesen Sie unter anderem

Simulationen auf dem ST Unwirkliche Welten

Seit Jahren versucht der Mensch, Naturvorgänge nachzuahmen. Gefährliche Vorgänge werden auf dem Rechner ausprobiert, ohne daß jemand etwas passieren kann oder sich eine Naturkatastrophe ereignet. Auf dem Computer kann auch ein Kind bis zum Mond fliegen oder ein Chemiker ein neues Präparat entwickeln, ohne daß das Labor gleich in die Luft geht. Aber wo sind die Grenzen? Ist alles zu simulieren? Ist der Rechner von heute die Werkstatt der Zukunft?

Turbo ST - Der Softwareblitter

Für alle, die bis jetzt noch auf einen kostengünstigen Blitter warten, bietet sich eine Softwarelösung für die beschleunigte Textausgabe an. Turbo ST soll angeblich sogar teilweise noch schneller als der Blitter arbeiten. Wir unterzogen das Produkt einem ausführlichen Test.

Soundmachine

Als Nachfolger für das erfolgreiche MusiX32 kam jetzt von Tommy Software ein neues Produkt namens Soundmachine ST auf den Markt. Einen ersten Vorgeschmack auf das, was einen erwartet, konnte man schon auf der ATARI-Messe in Düsseldorf bekommen. Im Gegensatz zu MusiX32 hat sich der Sound deutlich hörbar verbessert. Hörenswertes in der nächsten Ausgabe.

LCD-Overhead-Display von Crystal Vision

Ebenfalls von Messen bekannt ist das LCD-Display von Crystal Vision. Uns stand nun eine sehr interessante Variante, ein LCD-Display für Overheadprojektoren, zur Verfügung. Für Schulungszwecke dürfte es von großem Vorteil sein, da die Qualität doch deutlich über den bisher zu erhaltenden Produkten liegt.

Änderungen vorbehalten!

Die ST Computer-Ausgabe 2 '89 erscheint am 27.01.1989

Fragen an die Redaktion

Ein Magazin wie die ST-Computer zu erstellen, kostet sehr viel Zeit und Mühe. Da wir ja weiterhin vorhaben, die Qualität zu steigern (ja, auch wenn das manchmal danebengeht), haben wir Redakteure ein großes Anliegen an Sie, liebe Leserinnen und Leser:

Bitte haben Sie Verständnis dafür, daß Fragen an die Redaktion nur Donnerstags von 14[00]-17[00] Uhr telefonisch beantwortet werden können.

Vielen Dank für Ihr Verständnis

Impressum ST Computer

Chefredakteur:
Uwe Bärtels (UB)

Redaktion:
Uwe Bärtels (UB)
Harald Egel (HE)
Marcelo Merino (MM)
Harald Schneider (HS)

Redaktionelle Mitarbeiter:
C.Borgmeier (CBO)
Fernando Brand (FB)
Claus Brod (CB)
Stefan Höhn (SH)
Raymund Hofmann (RH)
Oliver Joppich (OJO)
Jürgen Leonhard (JL)
Claus P. Lippert (CPL)
Markus Nerding (MN)
Werner Schiewitz (WS)
Chr. Schormann (CS)
R.Tolksdorf (RT)

Autoren dieser Ausgabe:
A.Beller
J.Drücker
H.Katzenmayer
H.U.Mayer
R.Peiler
D.Rabich
Dr.K.Sarnow
H.D.Siewert
M.Wielebinski

Auslandskorrespondenz:
C.P.Lippert (Leitung)
D.dela Fuente (UK)
L.Hennelly (Nordamerika)

Redaktion: "MAXON" Computer GmbH
Postfach 59 69
Industriestr. 26
6236 Eschborn
Tel.: 0 61 96/48 18 11
FAX : 0 61 96/4 11 37

Verlag:
Heim Fachverlag
Heidelberger Landstr. 194
6100 Darmstadt 13
Tel.: 0 61 51/5 60 57
FAX : 0 61 51/5 56 89 + 5 60 59

Verlagsleitung:
H.J.Heim

Anzeigenverkaufsleitung:
U.Heim

Anzeigenverkauf:
K.Margaritis

Anzeigenpreise:
nach Preisliste Nr.3, gültig ab 1.1.88
ISSN 0932-0385

Grafische Gestaltung:
Bernd Weber, Kerstin Feist

Titelgestaltung:
Gunter Wenzel (Tel.: 06172/37193)

Fotografie:
Archiv, K.Ohlenschläger (Tel.: 06173/7400)

Produktion:
K.H.Hoffmann, B.Failer, S.Failer

Druck:
Ferling Druck GmbH

Lektorat:
V.Pfeiffer

Bezugsmöglichkeiten:
ATARI-Fachhandel, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag

ST Computer erscheint 11 x im Jahr
Einzelpreis: DM 7,-, ÖS 56,-, SFr 7,-
Jahresabonnement: DM 70,-
Europ. Ausland : DM 90,- Luftpost : DM 120,-
In den Preisen sind die gesetzliche MWSt. und die Zustellgebühren enthalten.

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit seiner Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern dem Heim Verlag.
Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.